Manipulating Structure in Images and Videos

Sagie Benaim

School of Computer Science, Tel Aviv University

TEL AVIV UNIVERSITY

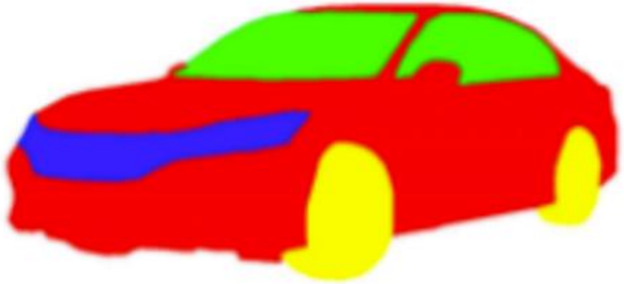
What is a natural image?

Texture

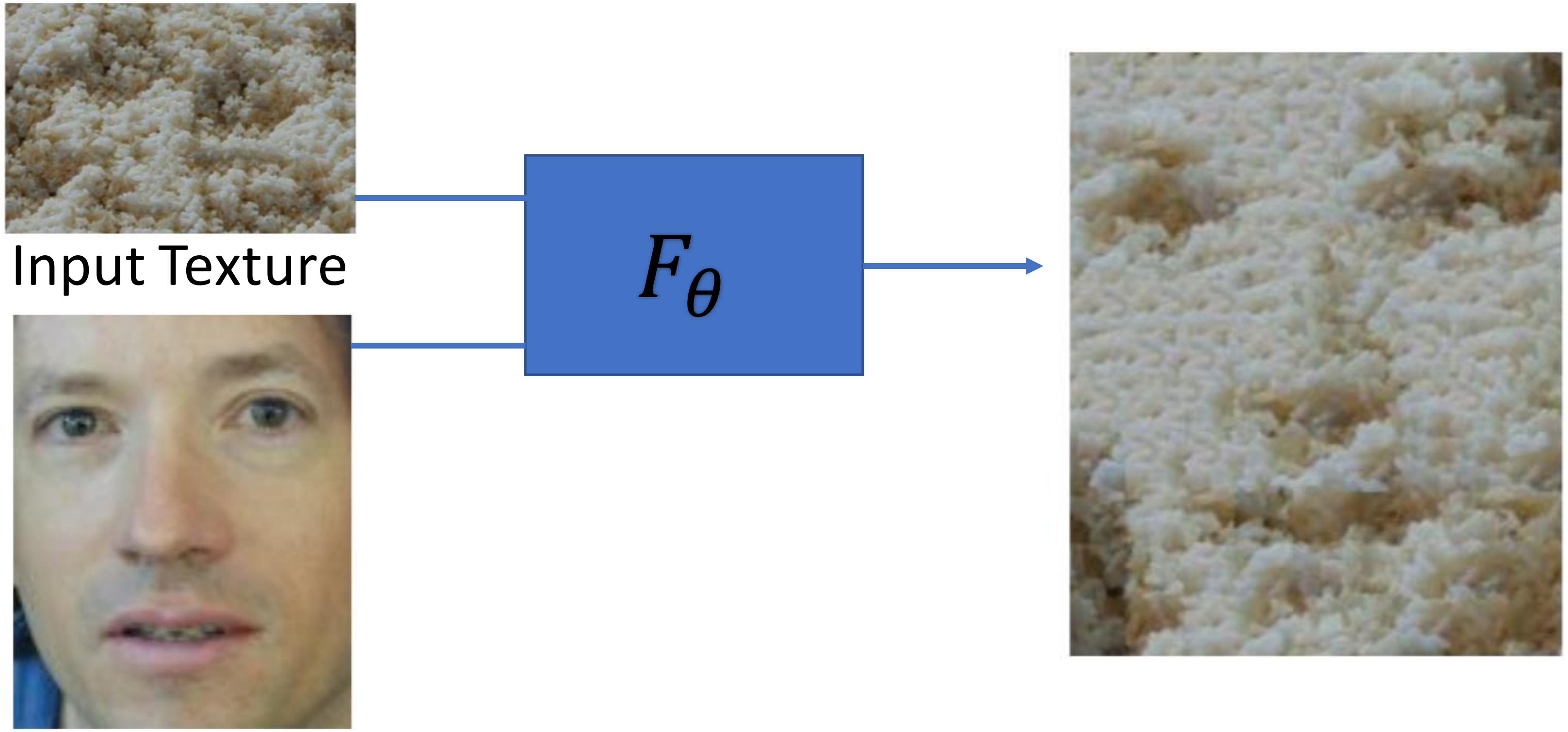
Style

L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style”. 2015.

Structure

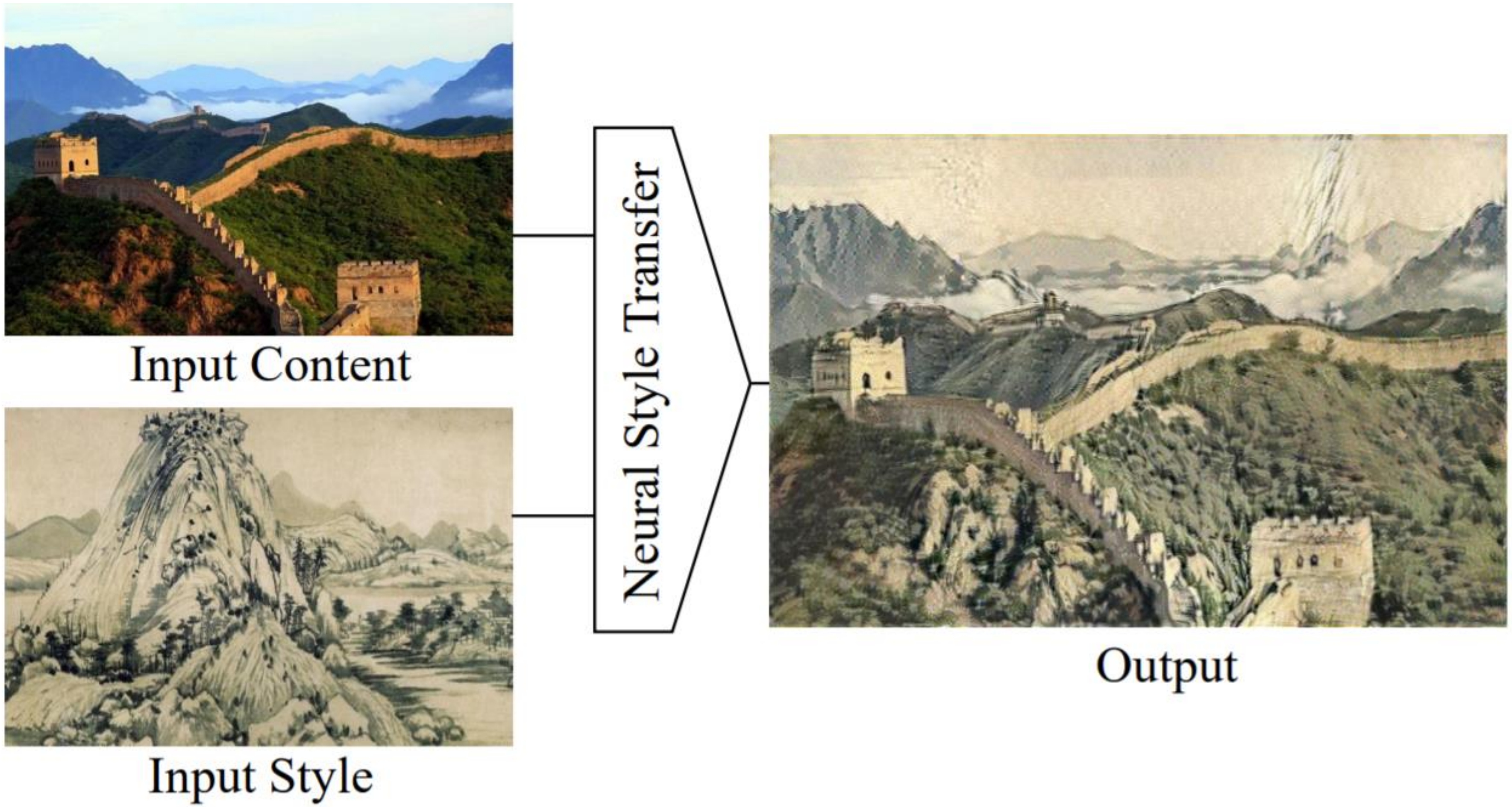

Manipulating Texture



Input Content

A.A.Efros, W.T.Freeman; “Image Quilting for Texture Synthesis and Transfer”; SIGGRAPH01

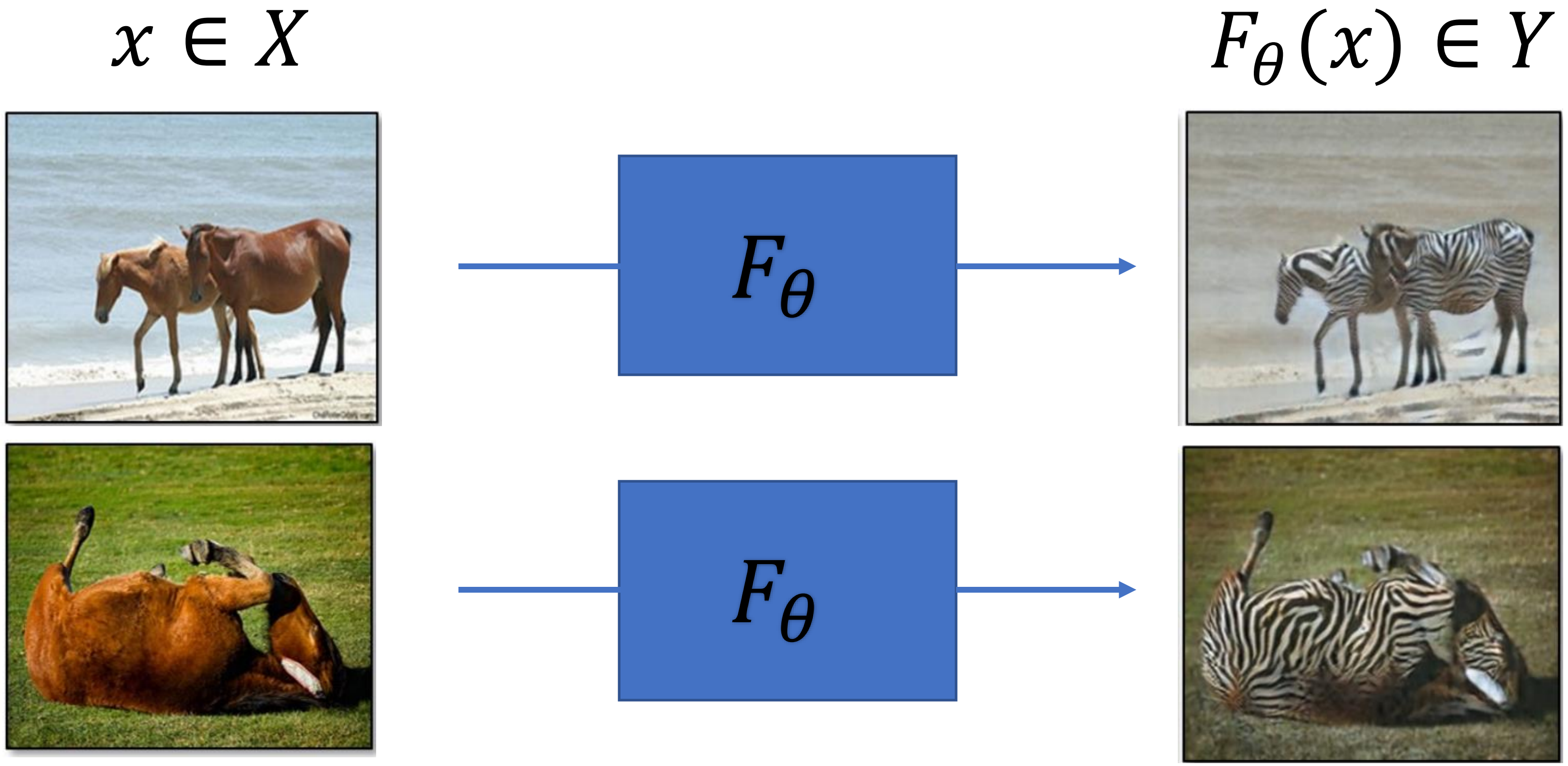
Manipulating Style



L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style”. 2015.

Image to Image Translation

1. $\mathrm{F}_\theta(\mathrm{x})$ preserves the **structure** of objects of x
2. $\mathrm{F}_\theta(\mathrm{x})$ belongs to Y's distribution (changes **style**)



CycleGAN, Zhu et al., ICCV 2017

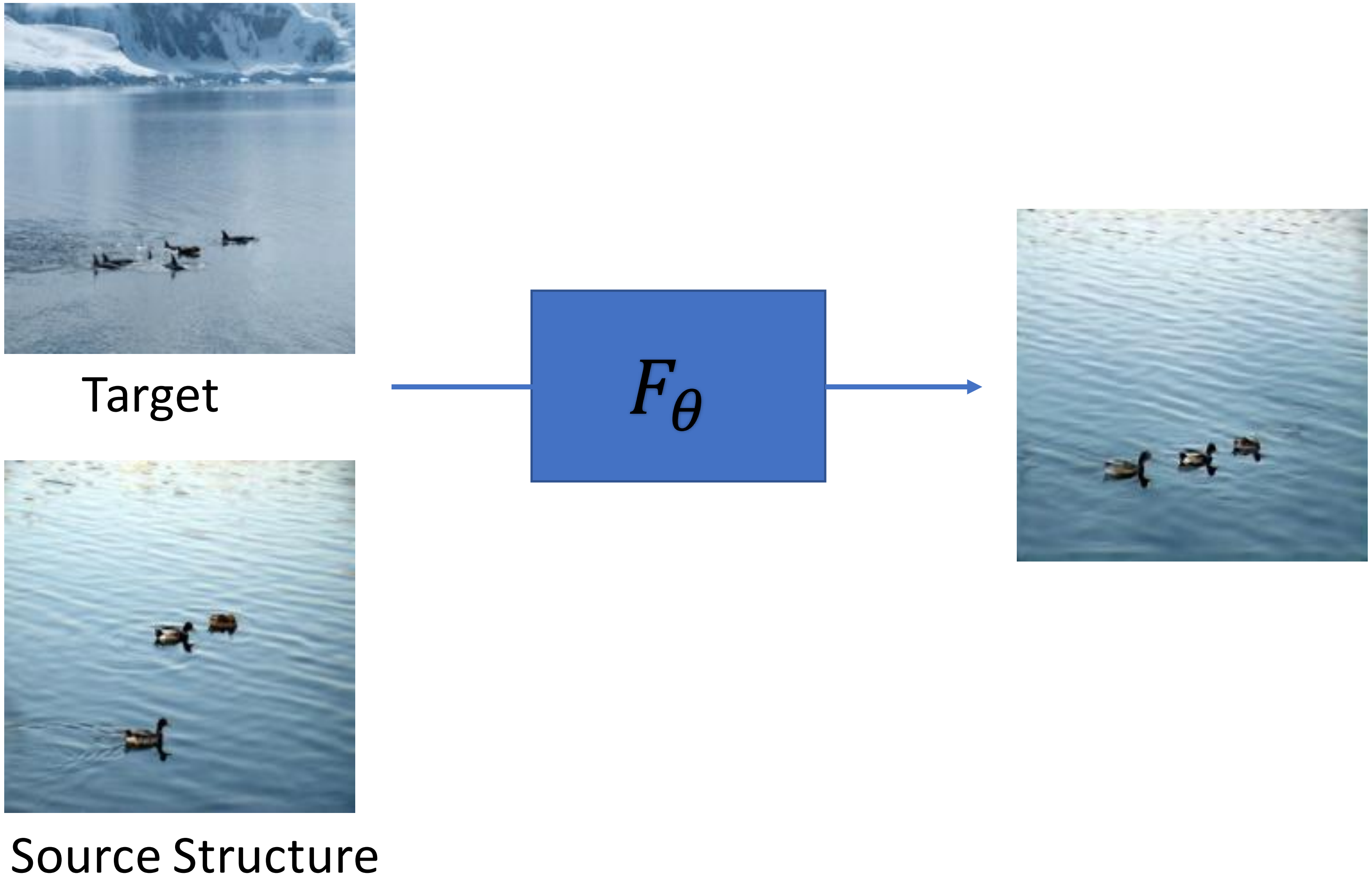
Manipulating Structure

Target

Source Structure

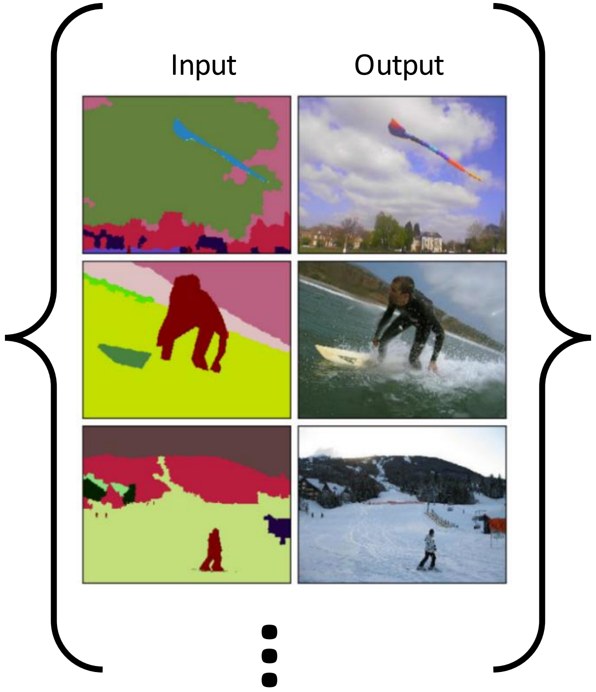
F_θ

Manipulating Structure

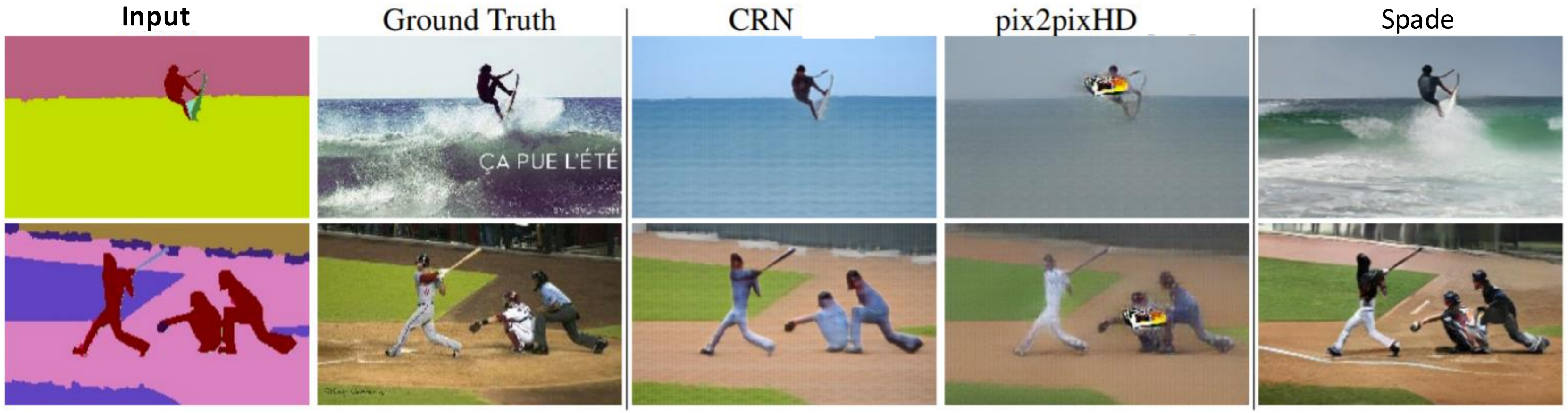


Supervised (Paired) Setting

Train



Test



Unsupervised (Unpaired) Setting

X

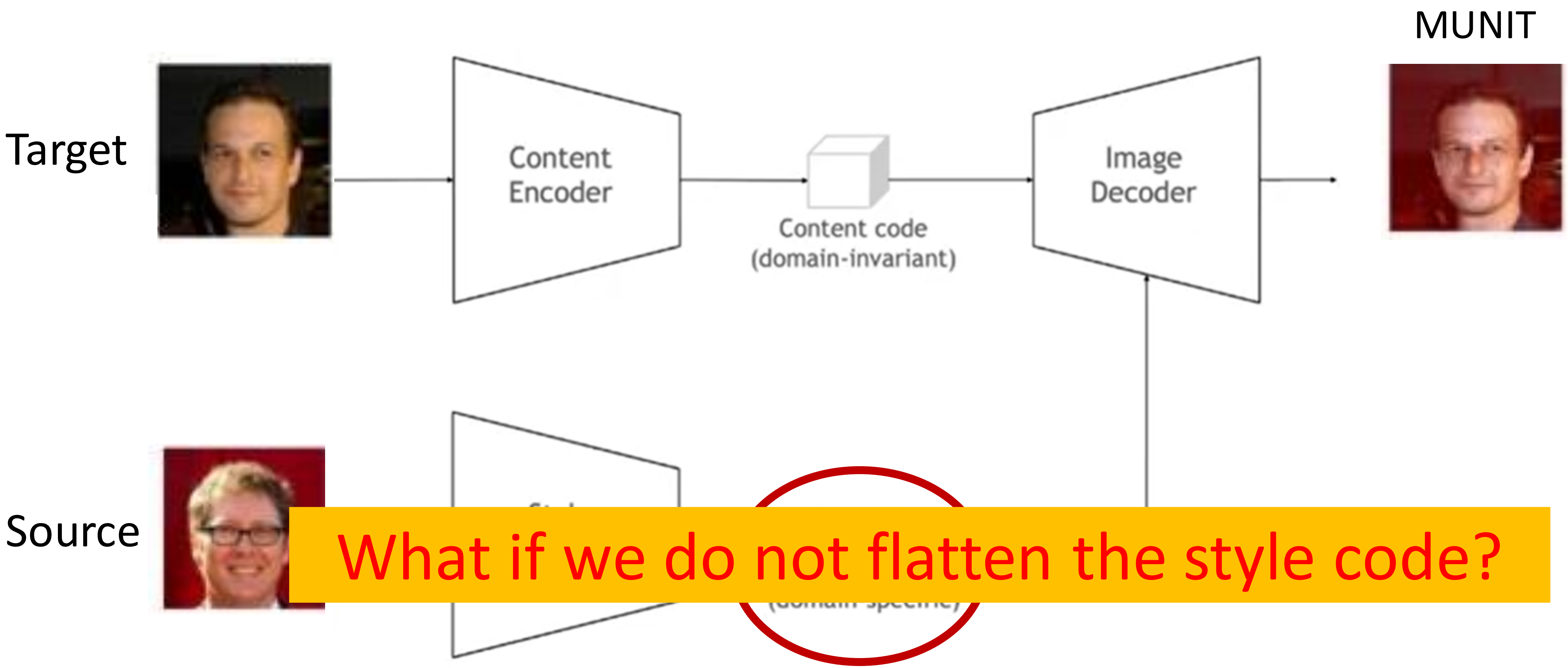
Faces <u>without</u> glasses

Y

Faces with glasses

Control Structure of Generated Faces (Transfer Glasses)

Multimodal Image to Image Translation



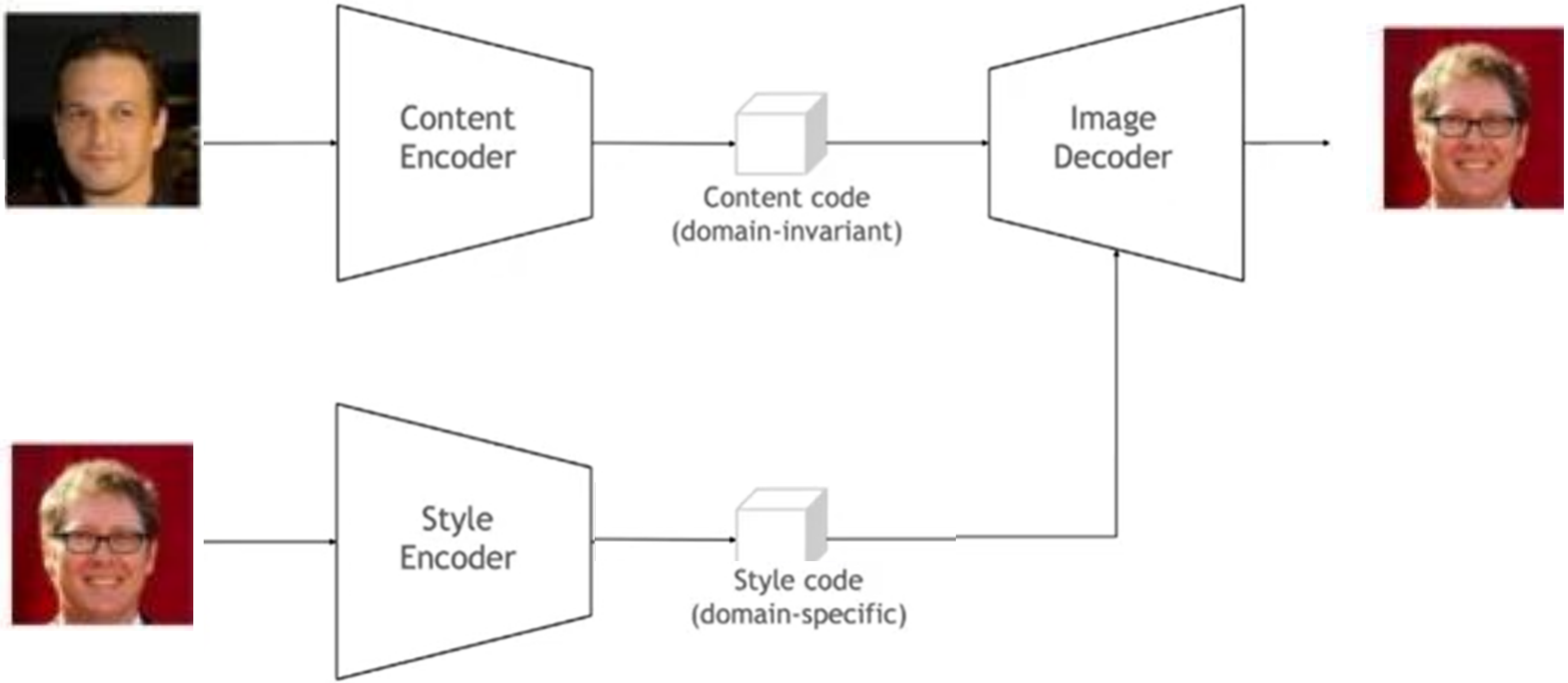
What if we do not flatten the style code?

Multimodal Image to Image Translation

MUNIT

Target

Source



Domain Intersection and Domain Difference

S. Benaim, M. Khaitov, T. Galanti, L. Wolf. ICCV 2019.

Given two visual domains, disentangle the **separate (domain specific)** information and **common (domain invariant)** information.

See also: Emerging Disentanglement in Auto-Encoder Based Unsupervised Image Content Transfer. ICLR 2019.
O. Press, T. Galanti, **S. Benaim**, L. Wolf

Unsupervised Content Transfer

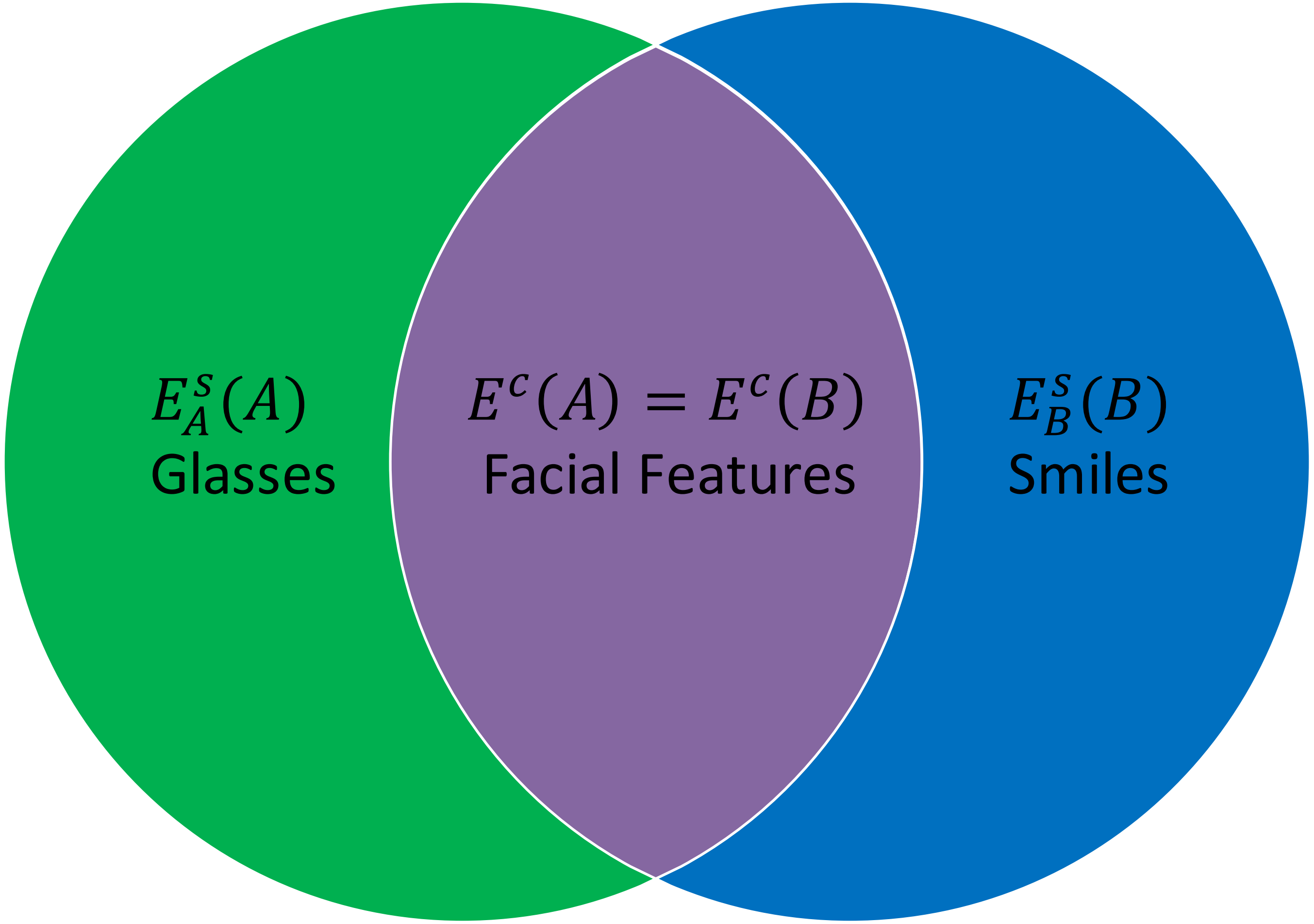
A

Non-smiling faces with glasses

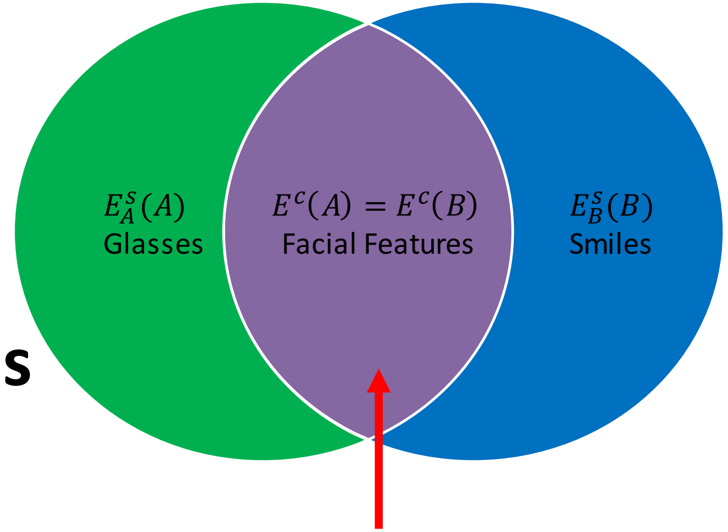
B

Smiling faces without glasses

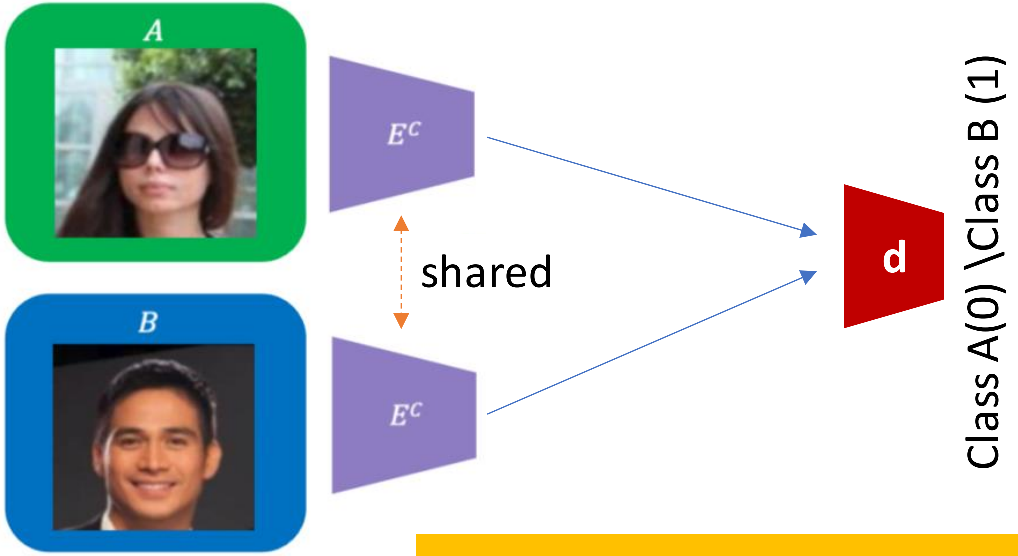
1. "Common" latent space, $E^c(A) = E^c(B)$. The space of **common facial features.**
2. "Separate" latent space for domain A, $E^s_A(A)$. The **space of glasses.**
3. "Separate" latent space for domain B, $E^s_B(B)$. The **space of smiles.**



The "common" Loss



Ensures E_c encodes information common to both domains



Discriminator d attempts to separate distributions (classify to correct label):

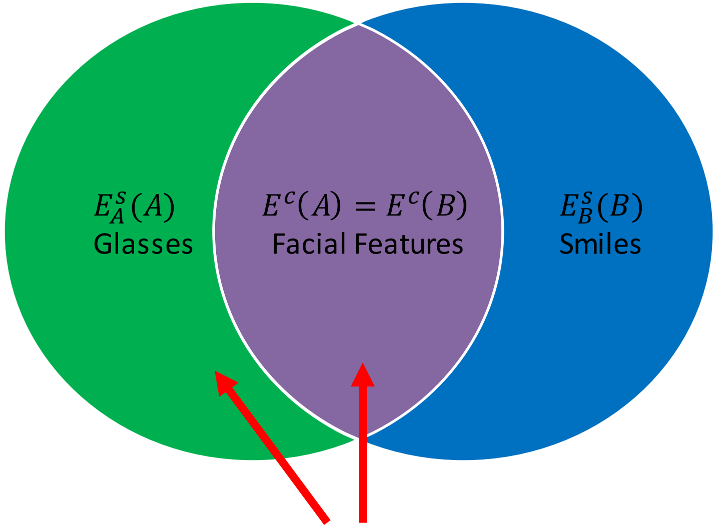
$$\frac{1}{m_1}\sum_{i=1}^{m_1} l(d(E^c(a_i)),0) + \frac{1}{m_2}\sum_{j=1}^{m_2} l(d(E^c(b_j)),1)$$

Encoder E_c attempts to match distributions of $E_c(A)$ and $E_c(B)$:

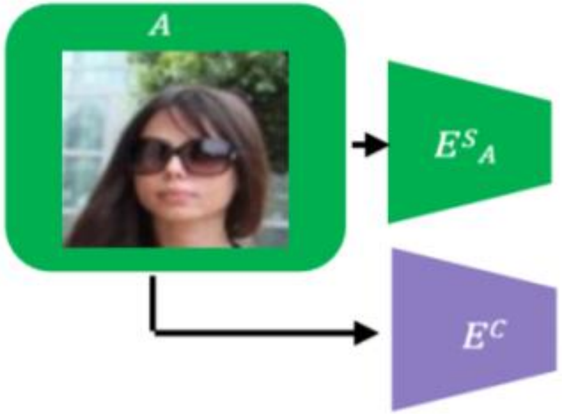
$$\frac{1}{m_1}\sum_{i=1}^{m_1} l(d(E^c(a_i)),1) + \frac{1}{m_2}\sum_{j=1}^{m_2} l(d(E^c(b_j)),1)$$

d can encode zero information

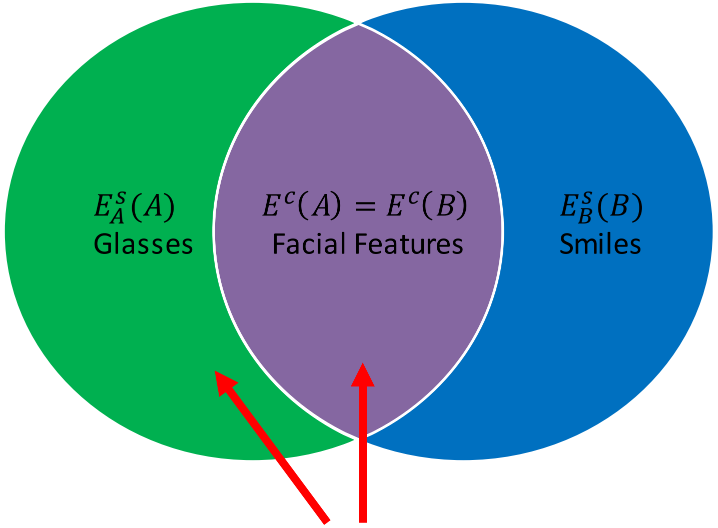
Reconstruction Losses



Ensures the "common" and "separate" encodings contain all the information in A

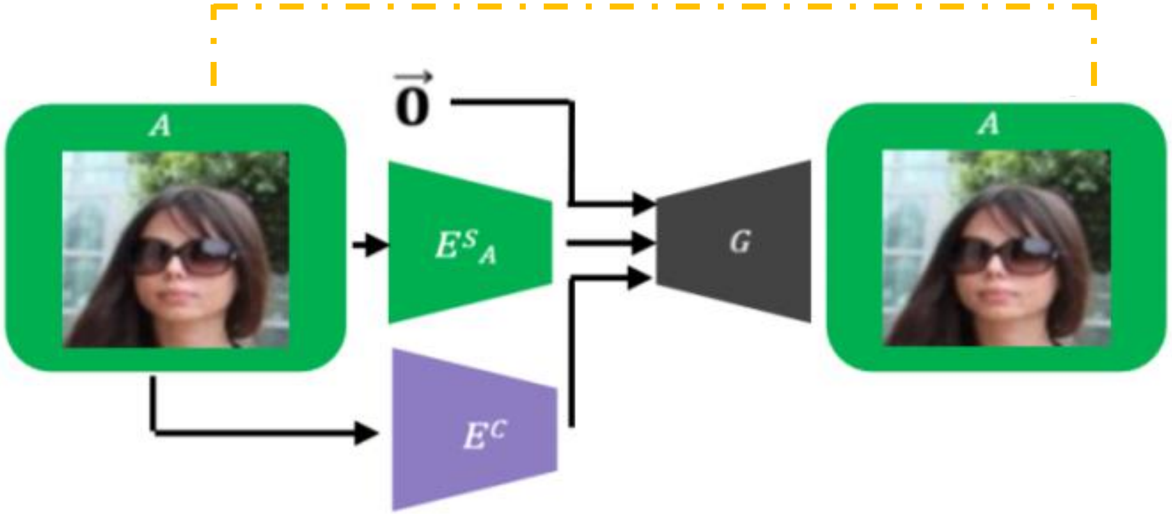


Reconstruction Losses

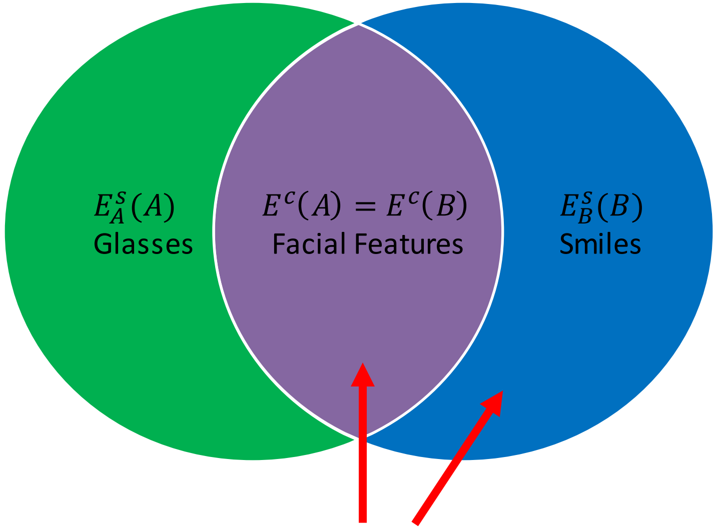


Ensures the "common" and "separate" encodings contain all the information in A

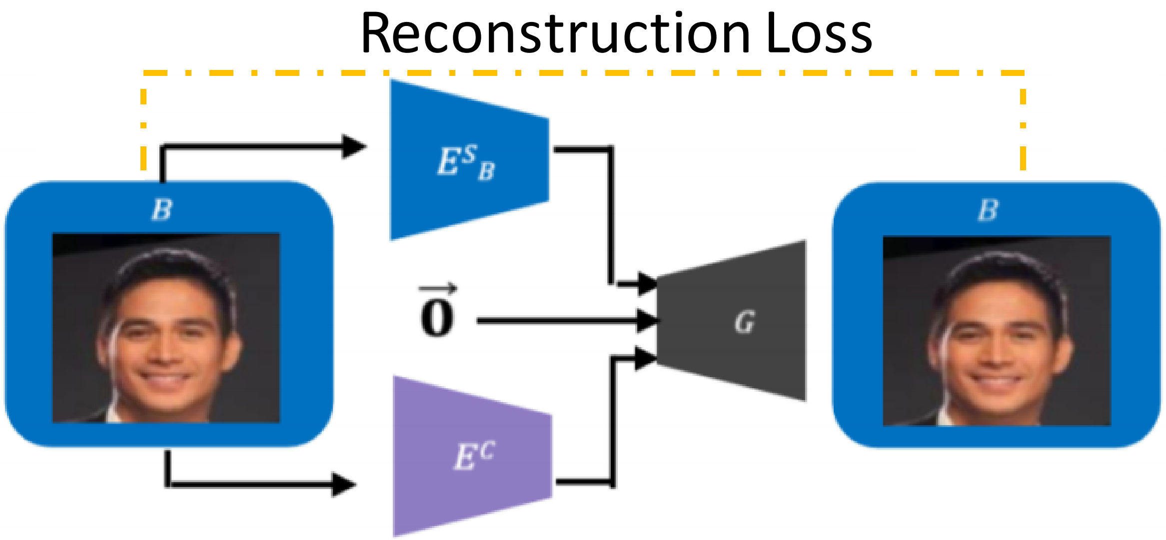
Reconstruction Loss



Reconstruction Losses

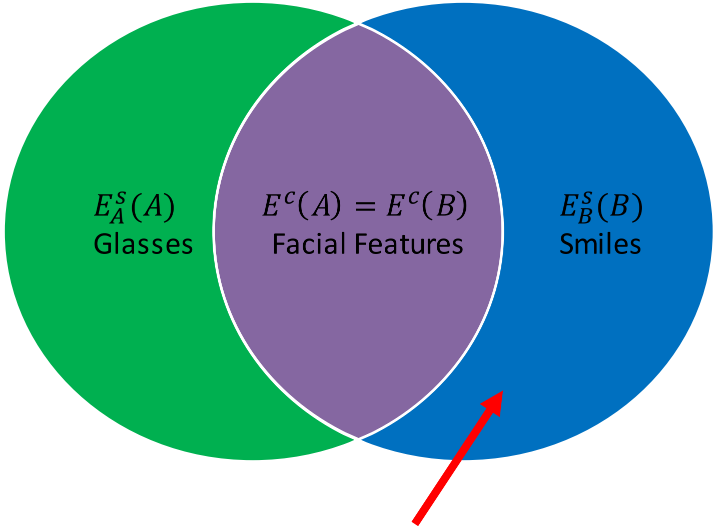


Ensures the "common" and "separate" encodings contain all the information in A



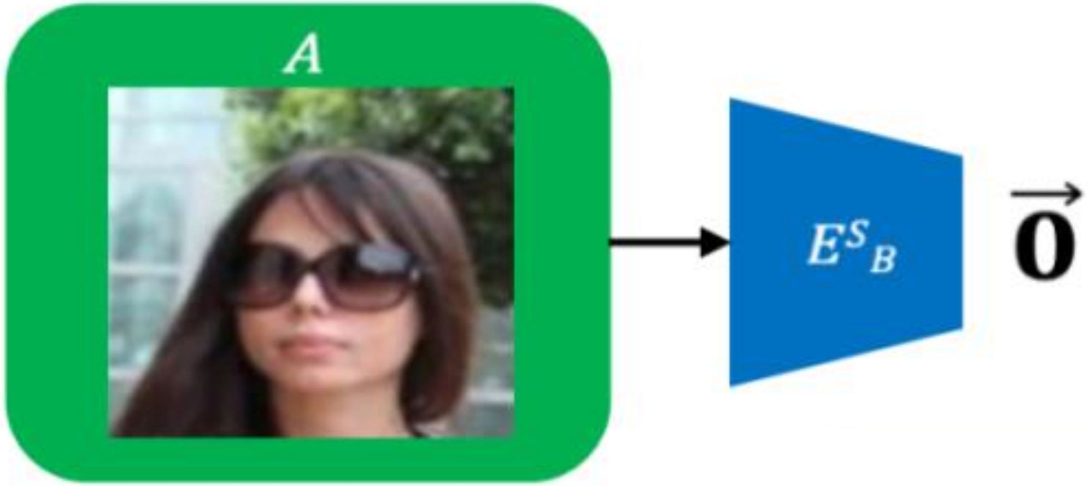
E_A^S (E_B^S)can encode all the information of A (B)

"Zero" Loss

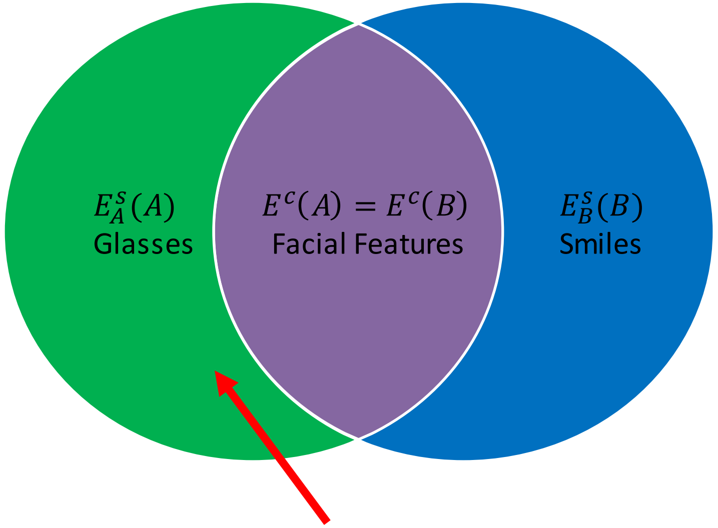


Ensures the separate encoder of B does not encode information about A

$$\mathcal{L}_{zero}^{B} := \frac{1}{m_1}\sum_{i=1}^{m_1} \|E_B^s(a_i)\|_1$$

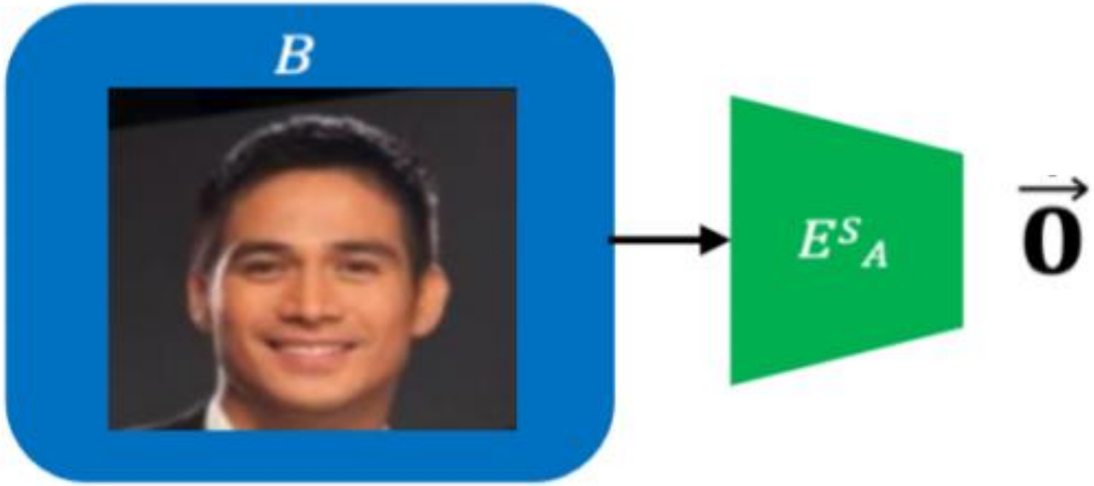


"Zero" Loss

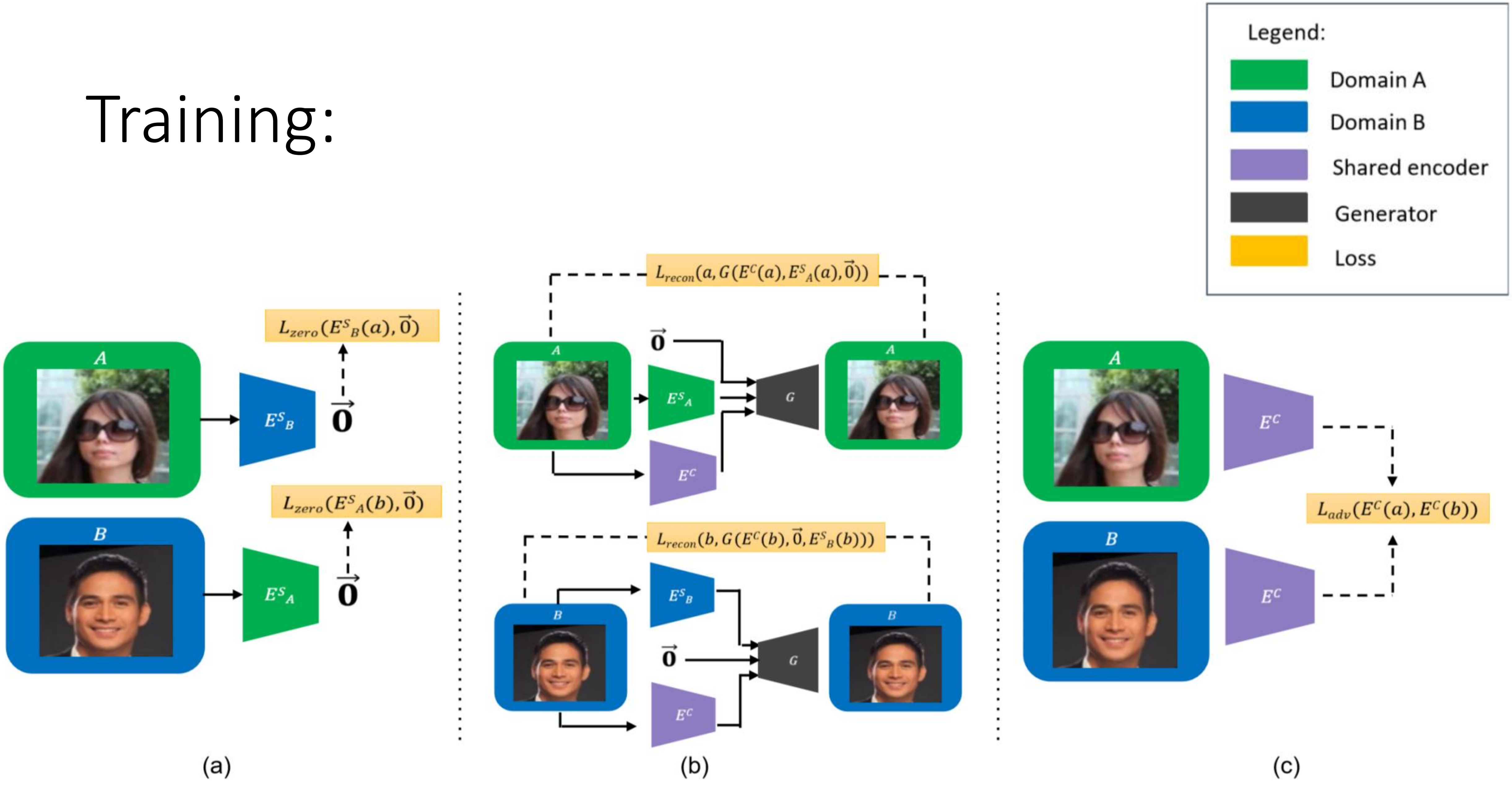


Ensures the separate encoder of B does not encode information about A

$$\mathcal{L}_{zero}^{A} := \frac{1}{m_2} \sum_{j=1}^{m_2} \|E_A^s(b_j)\|_1$$



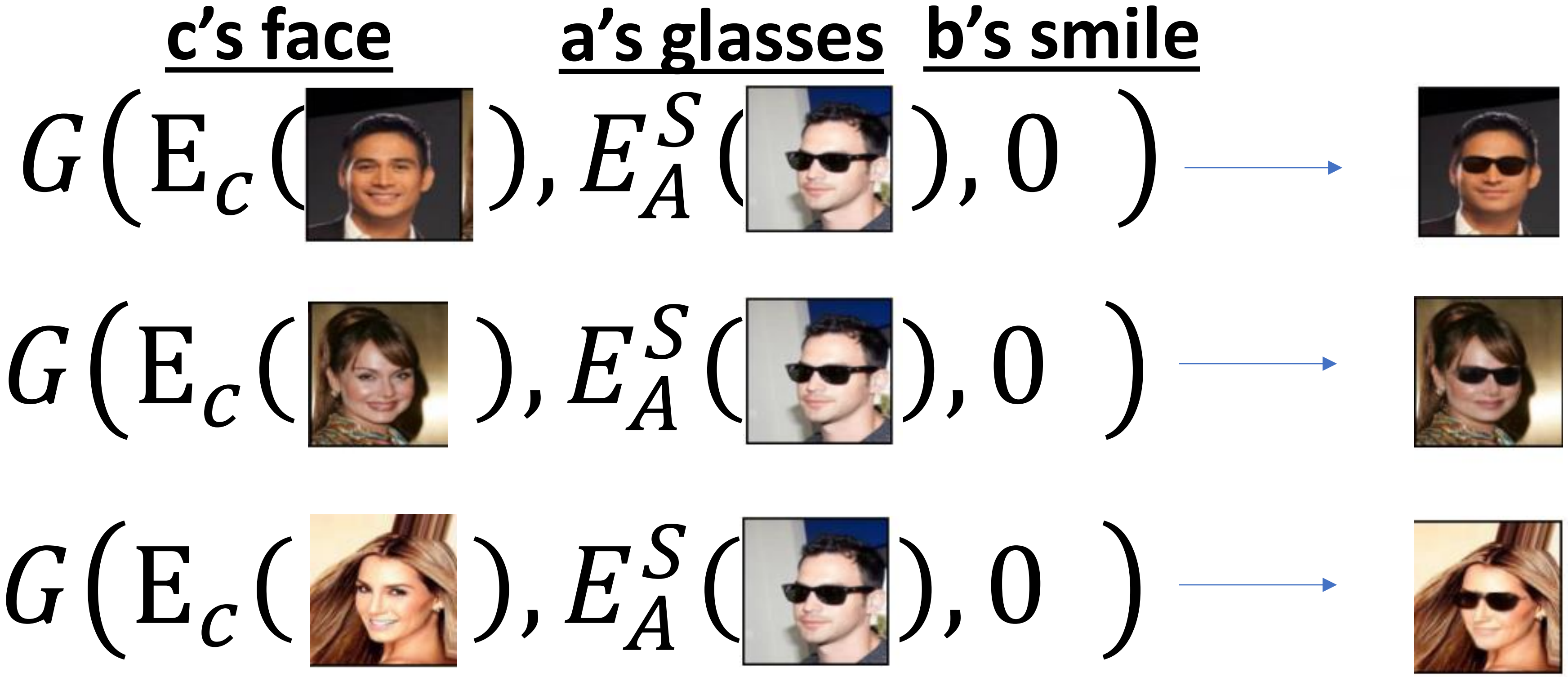
Training:



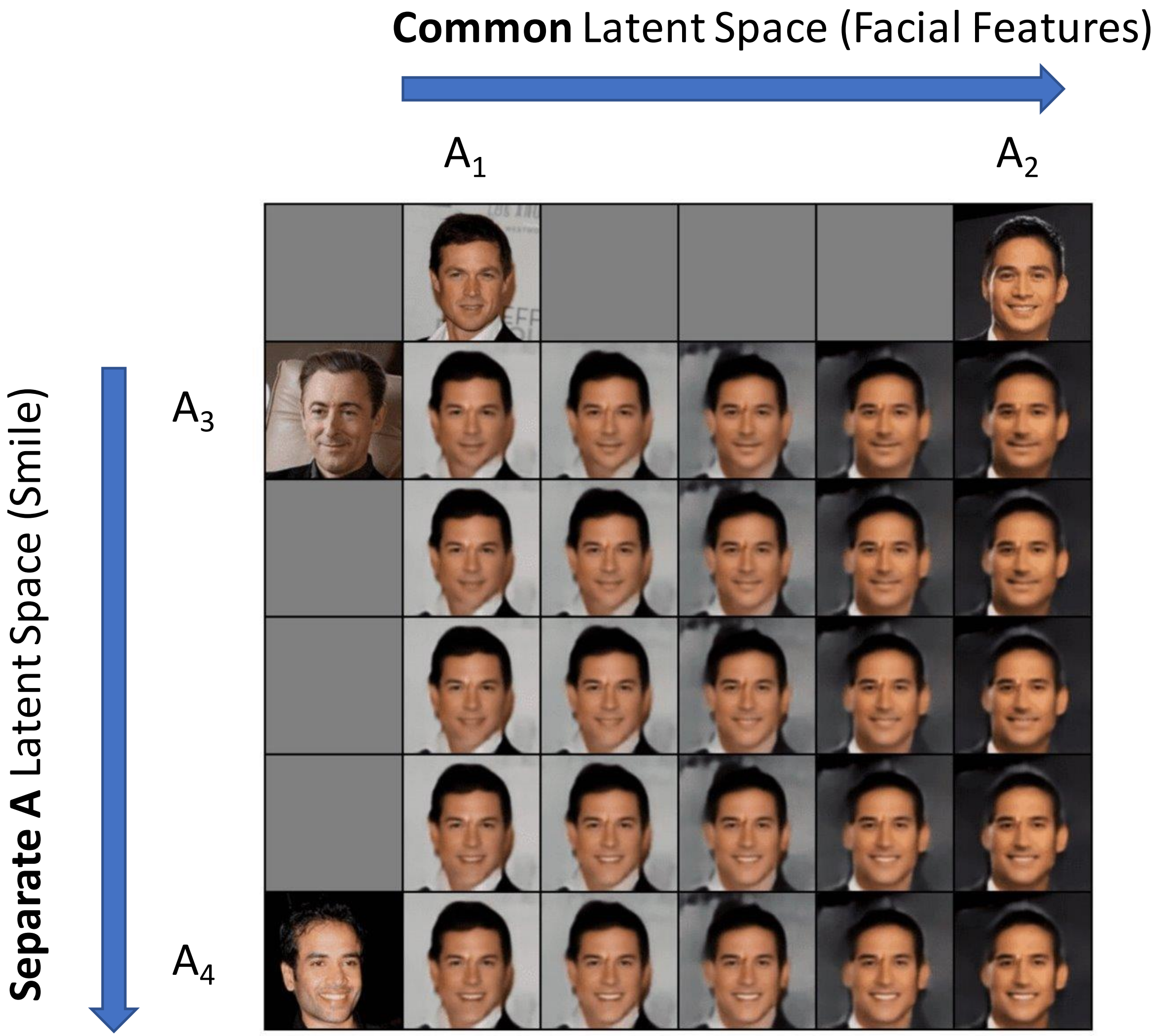
Legend:
Domain A
Domain B
Shared encoder
Generator
Loss
A
B
$E^S{}_B$
$E^S{}_A$
$\vec{0}$
$L_{zero}(E^S{}_B(a), \vec{0})$
$L_{zero}(E^S{}_A(b), \vec{0})$
$L_{recon}(a, G(E^C(a), E^S{}_A(a), \vec{0}))$
$L_{recon}(b, G(E^C(b), \vec{0}, E^S{}_B(b)))$
E^C
G
$L_{adv}(E^C(a), E^C(b))$
(a)
(b)
(c)

$$G\left(\mathrm{E}_c(c), E_A^S(a), E_B^S(b)\right) \longrightarrow$$

c's face
a's glasses
b's smile



Interpolation



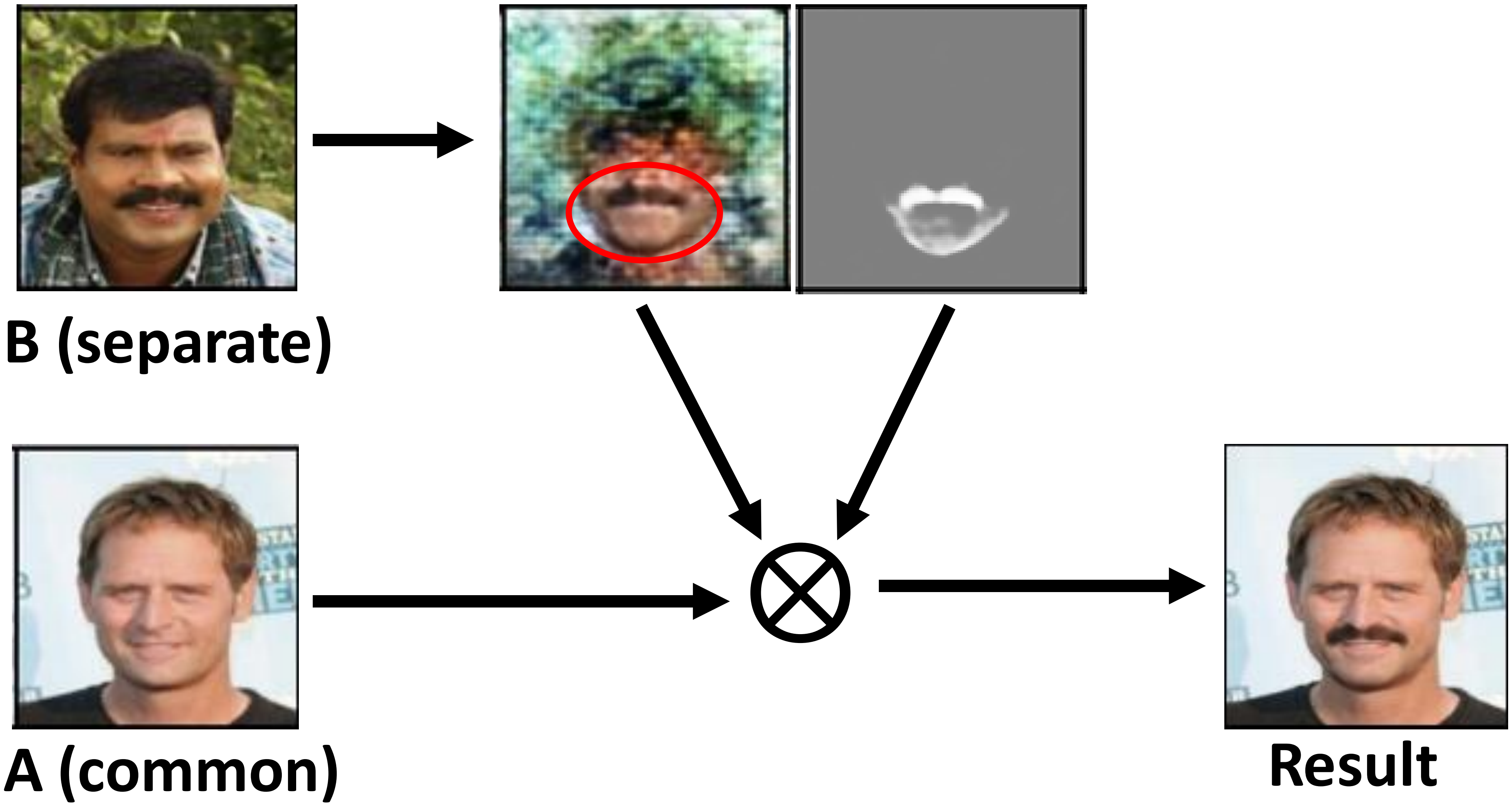
Losses “Necessary” and “Sufficient”

Under mild assumptions (such as our losses being minimized):

- $E^c(a)$ and $E_A^S(a)$ are independent (Similarly for B).
- $E^c(a)$ and $E_A^S(a)$ captures the true underlying “common” and “separate” information in a (Similarly for B).
- I.e., our losses are both **necessary and sufficient** for the desired **disentanglement**.

Masked Based Unsupervised Content Transfer

R. Mokady, **S. Benaim**, L. Wolf, A. Bermano. ICLR 2020.




Common
Source
Glasses
Separate

Two Attributes

Attribute removal

Input **Result**

Facial Hair Removal

Input **Result**

Smile Removal

Out of Domain Manipulation

Weakly-Supervised Segmentation

Table 5: Mean and SD IoU for the two hair segmentation benchmarks.

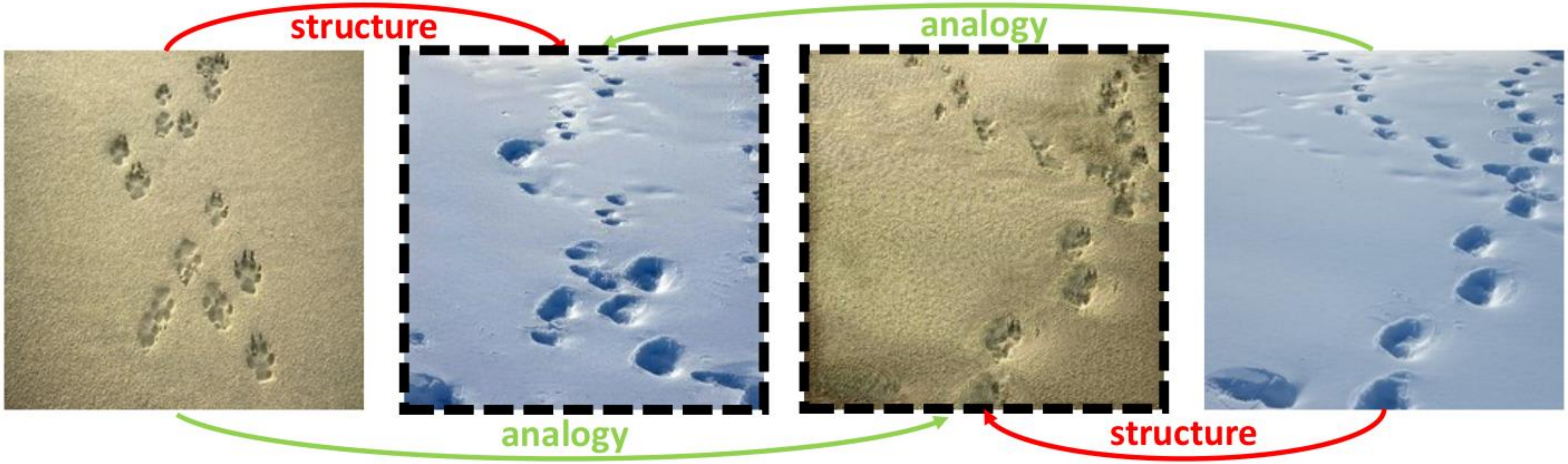
Method	Women's hair	Men's hair
Ours	0.77± 0.15	0.77 ± 0.13
Press et al.	0.67± 0.13	0.58 ± 0.11
Ahn & Kwak.	0.54± 0.10	0.52 ± 0.10
CAM	0.43± 0.09	0.56 ± 0.07

GT | Ours | Press et al. | Ahn et al. | CAM

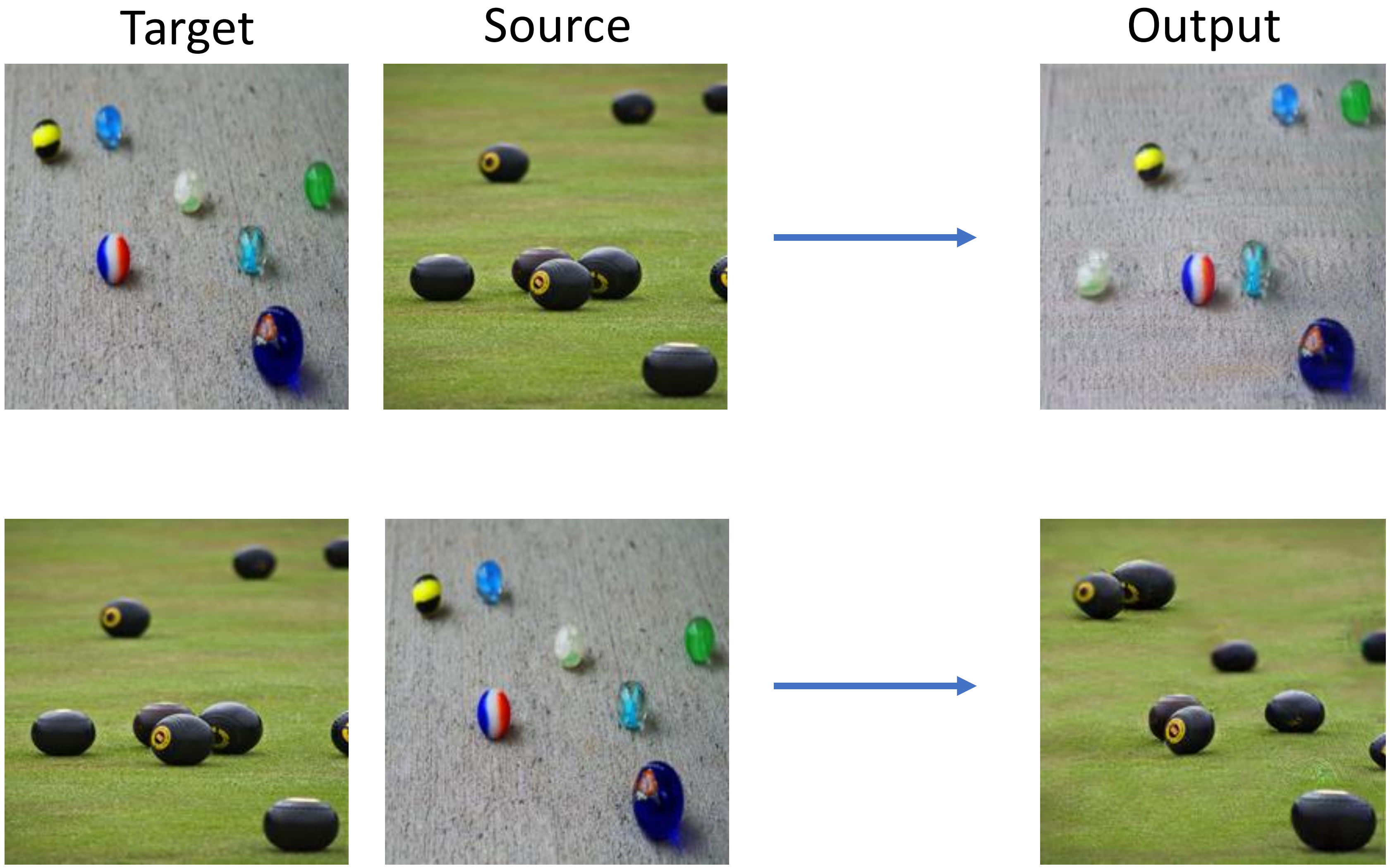
Structural-analogy from a **Single Image Pair**

S. Benaim*, R. Mokady*, A. Bermano, D Cohen-Or, L. Wolf. CGF 2020. (*Equal contribution)

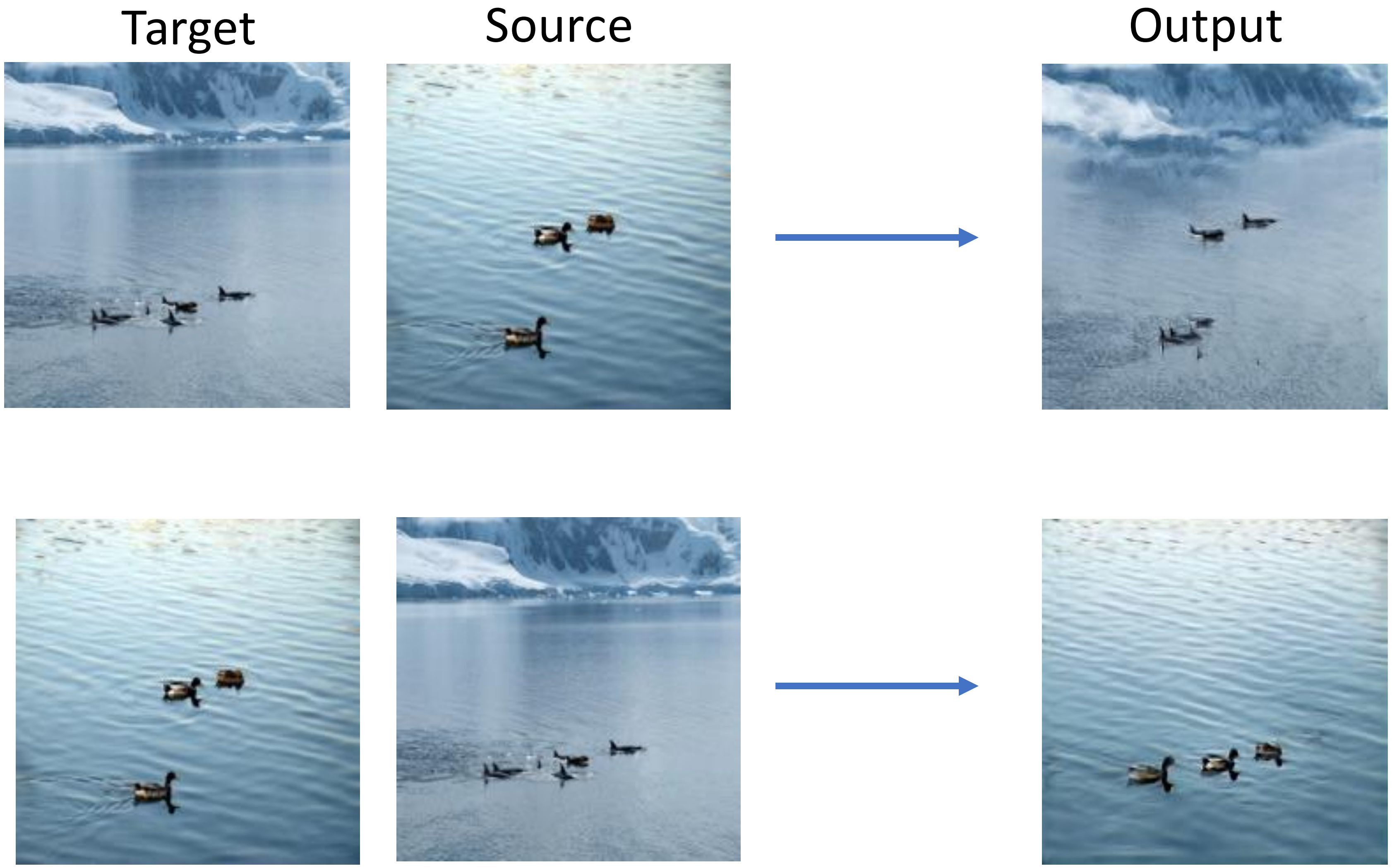
Generate an image which is **aligned to the source** image but depicts **structure from a target image**



Structural Analogy

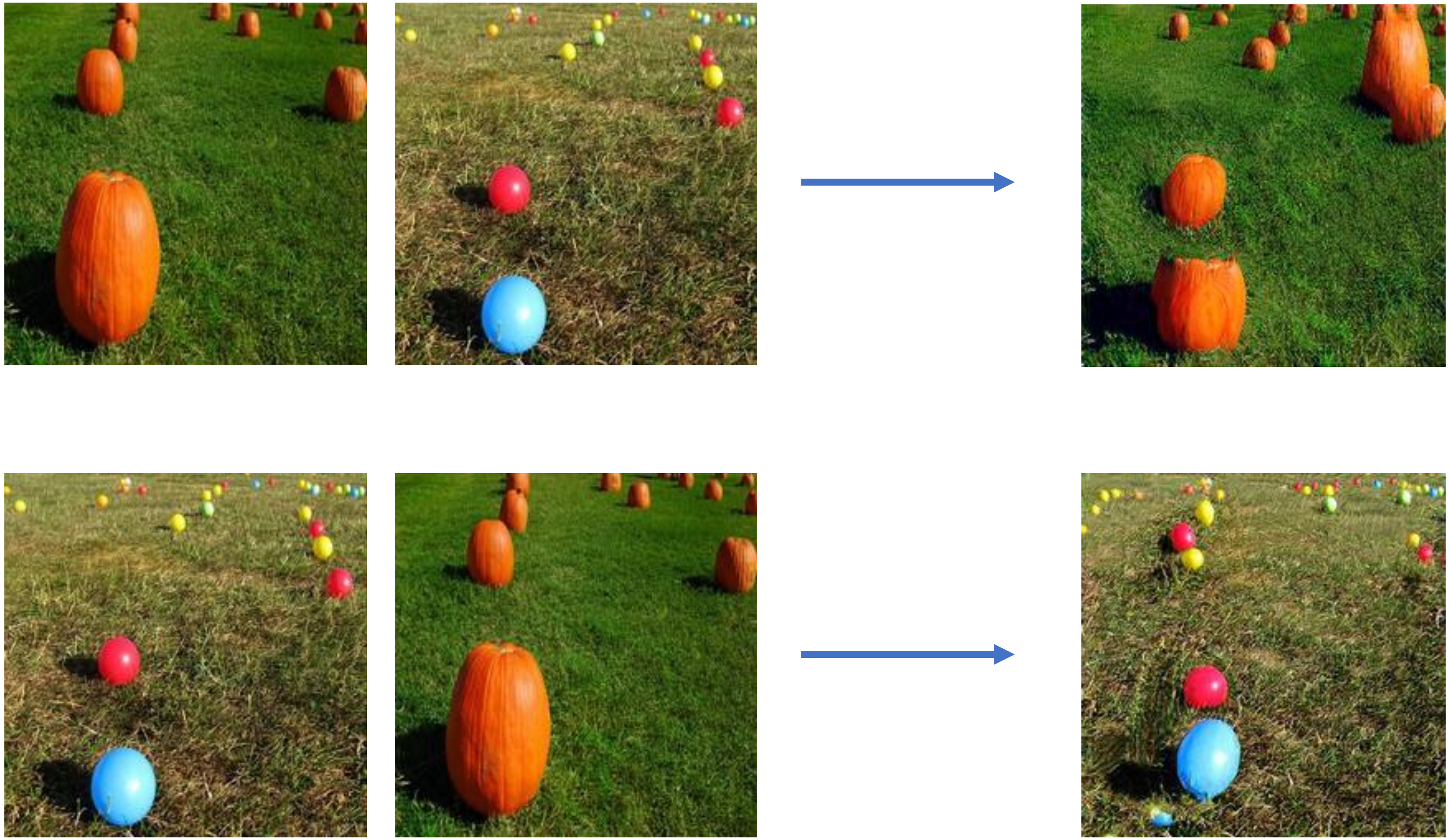


Structural Analogy



Structural Analogy

Target Source Output



Style Transfer

Style

Content

Result

Deep Image Analogy

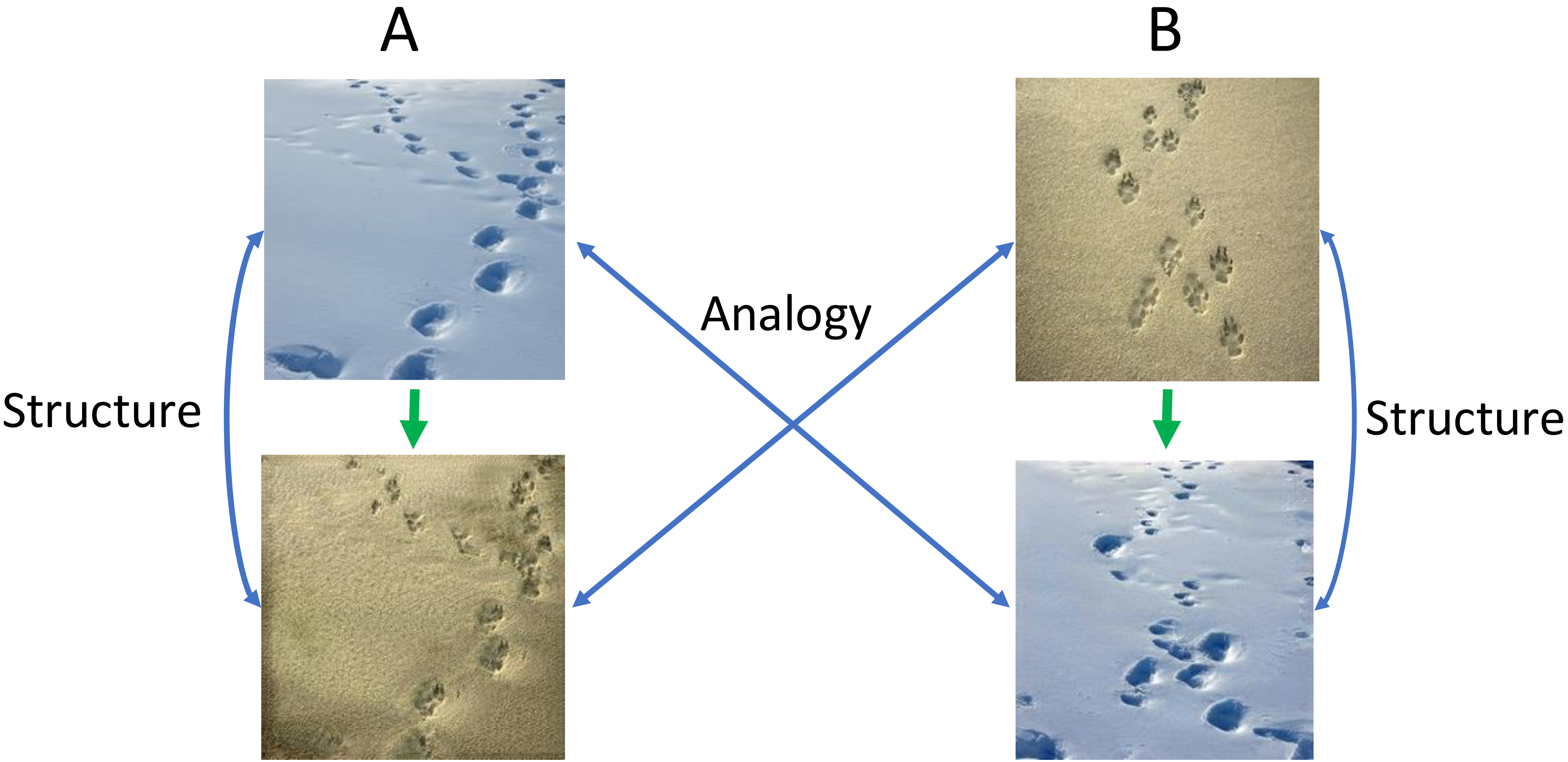
Style

Content

Result

Cannot Change Object Shape

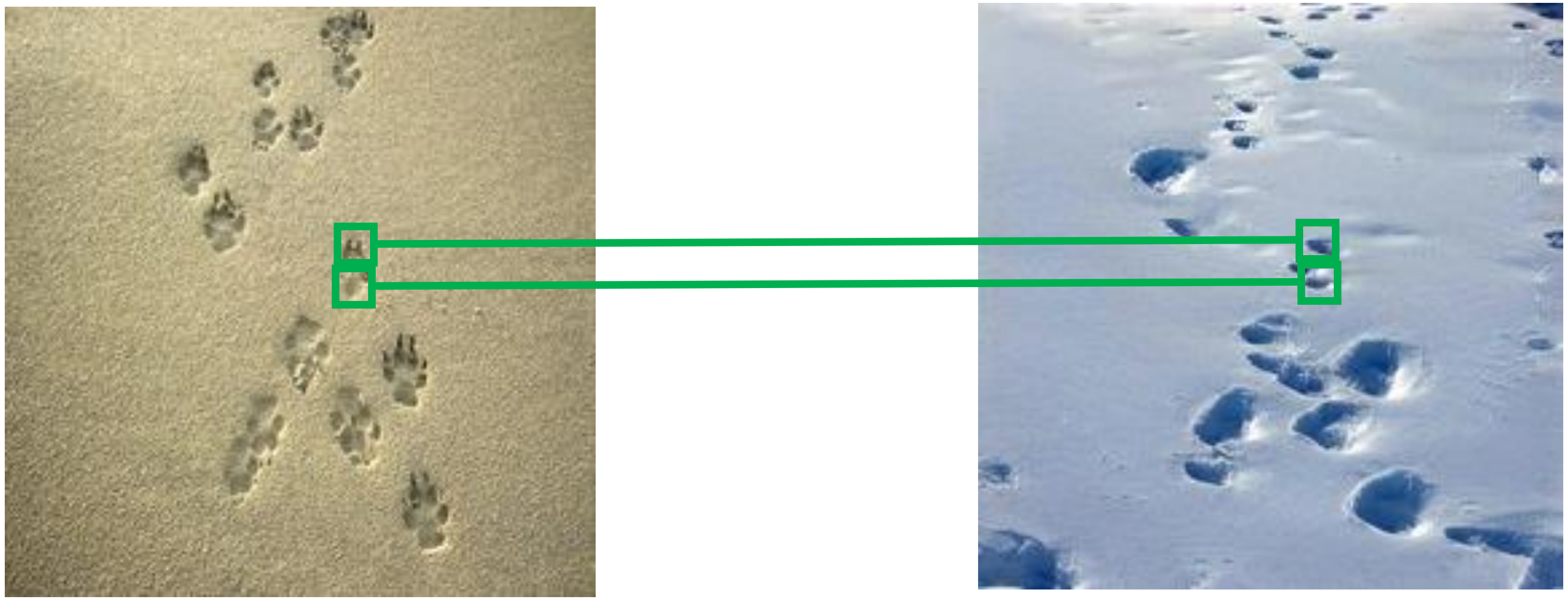
Structural Analogy



Motivation

Motivation

Motivation



Proposed Hierarchical Approach

Coarsest scale:
Large Patches

$\bar{a}^0$(Unconditional)
$\overline{ab}^0$(Conditional)

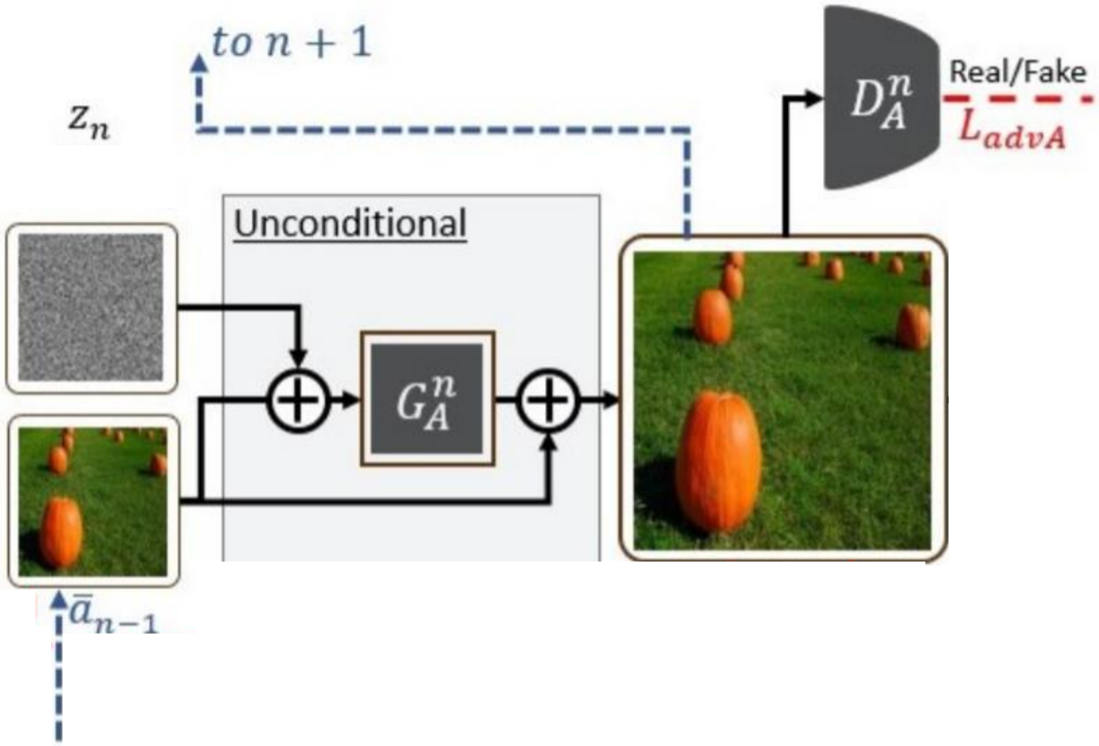
LEVEL $= 0$

Finest scale:
Small Patches

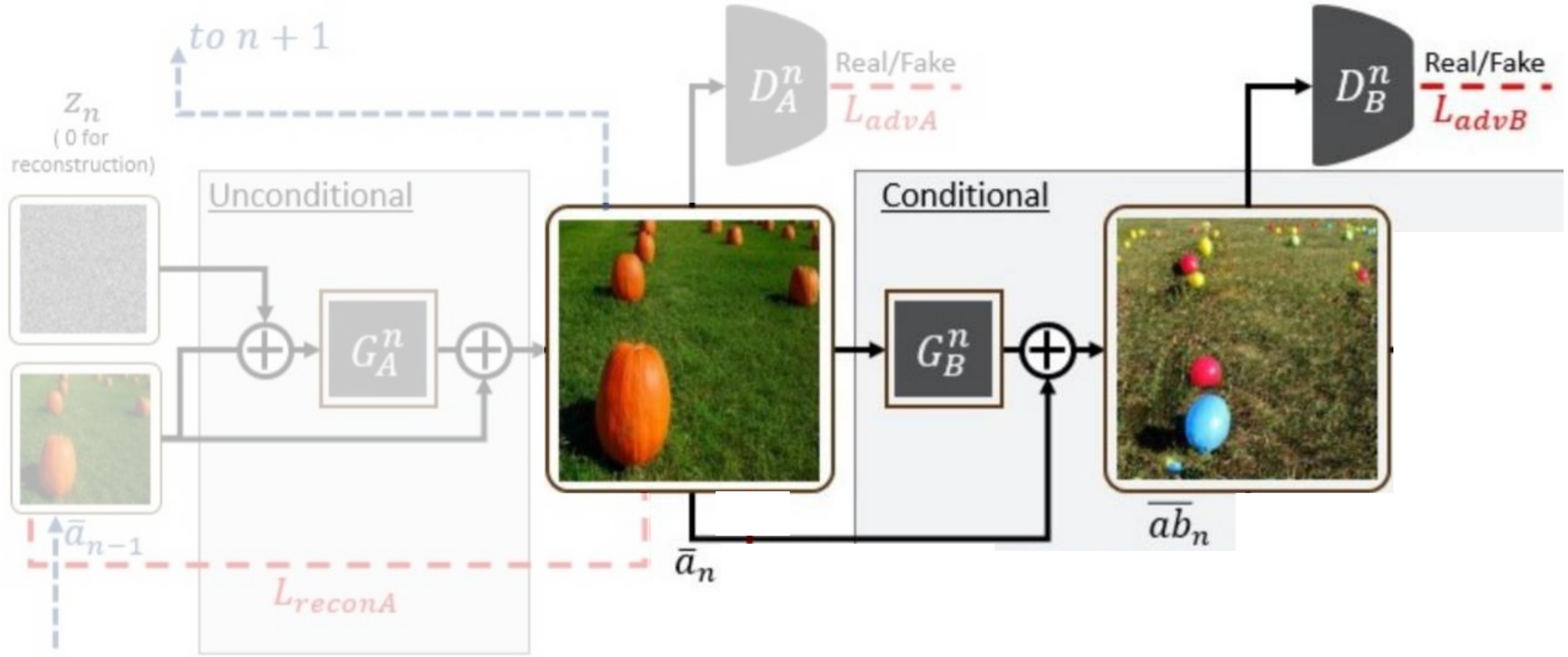
$\bar{a}^N$(Unconditional)
$\overline{ab}^N$(Conditional)

LEVEL $= N$

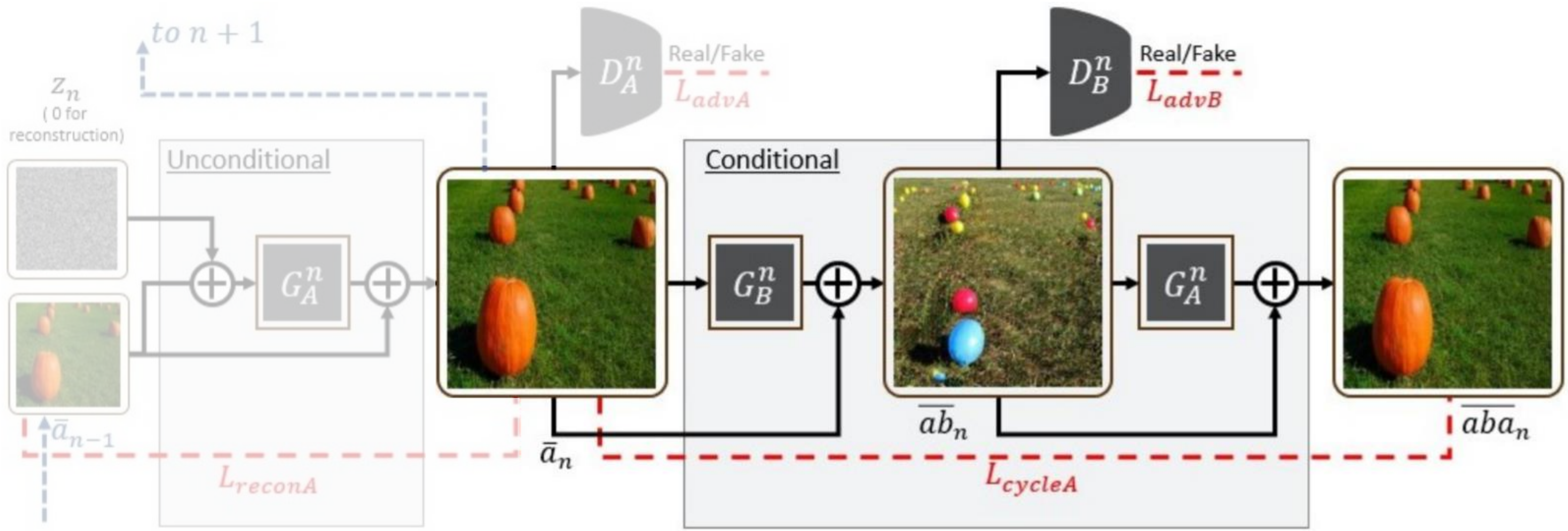
Unconditional Generation (Level n)



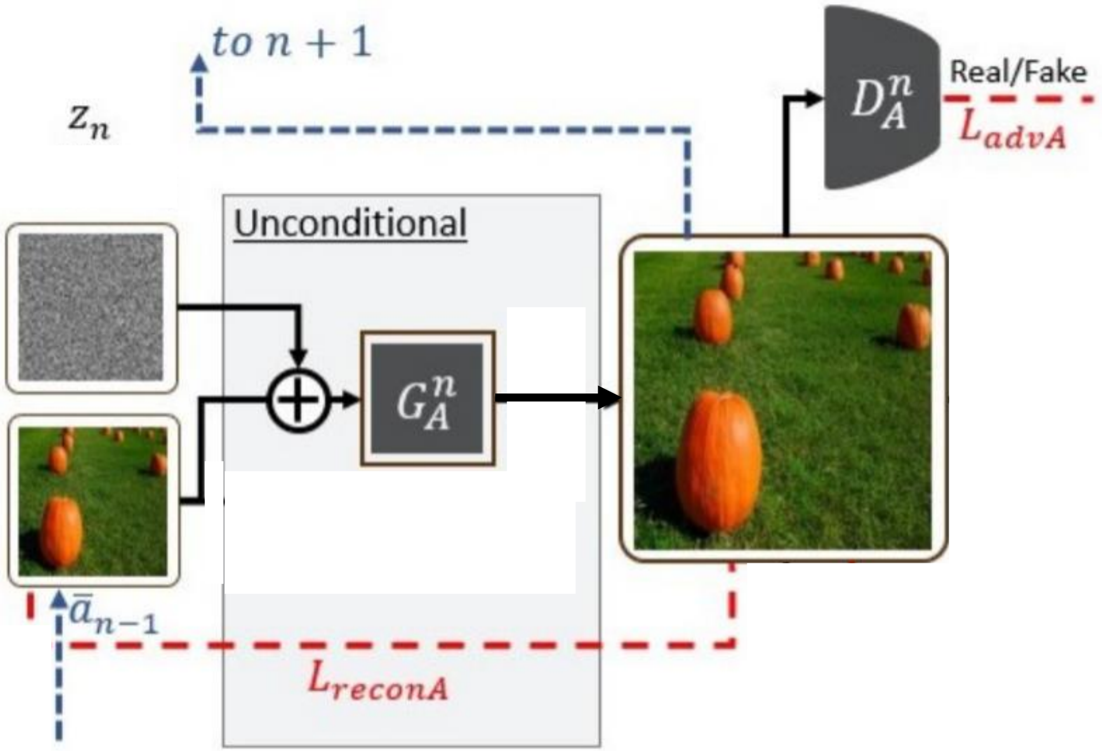
Conditional Generation (Level n)



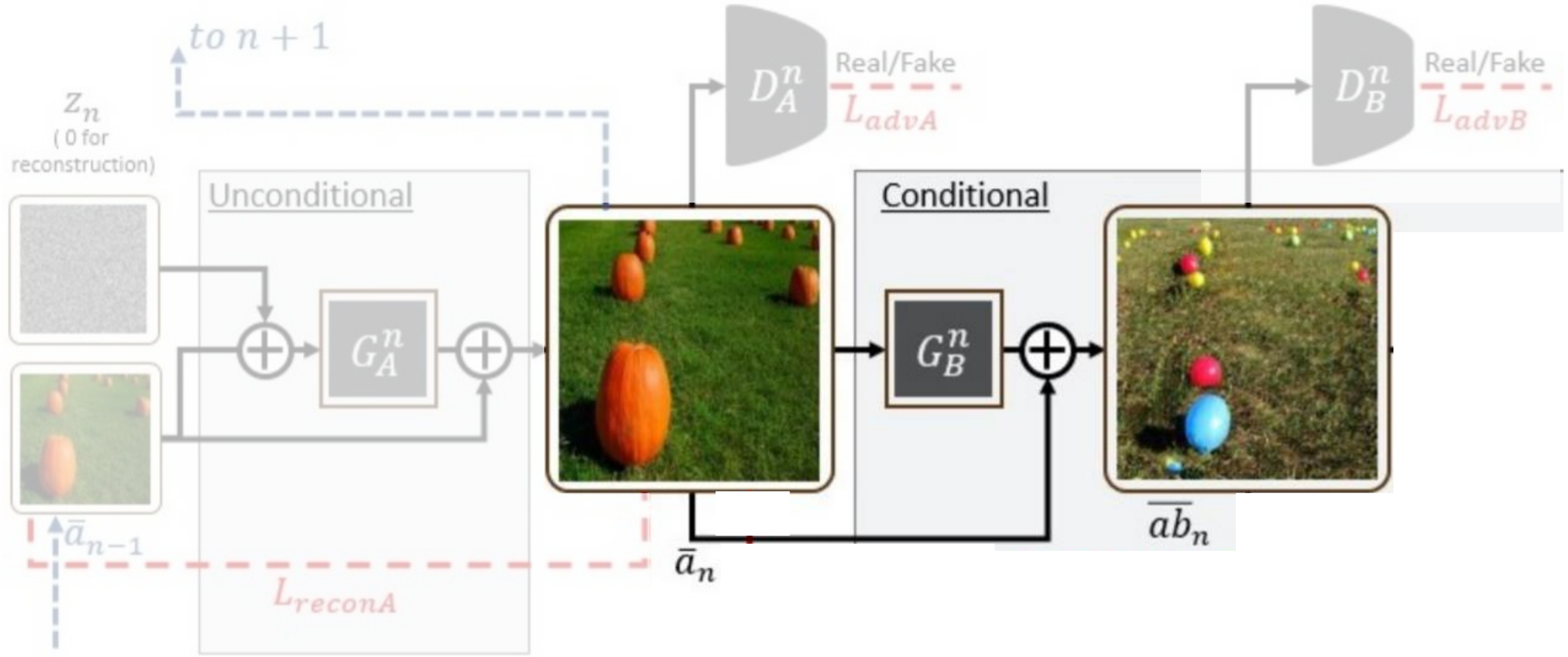
Conditional Generation (Level n)

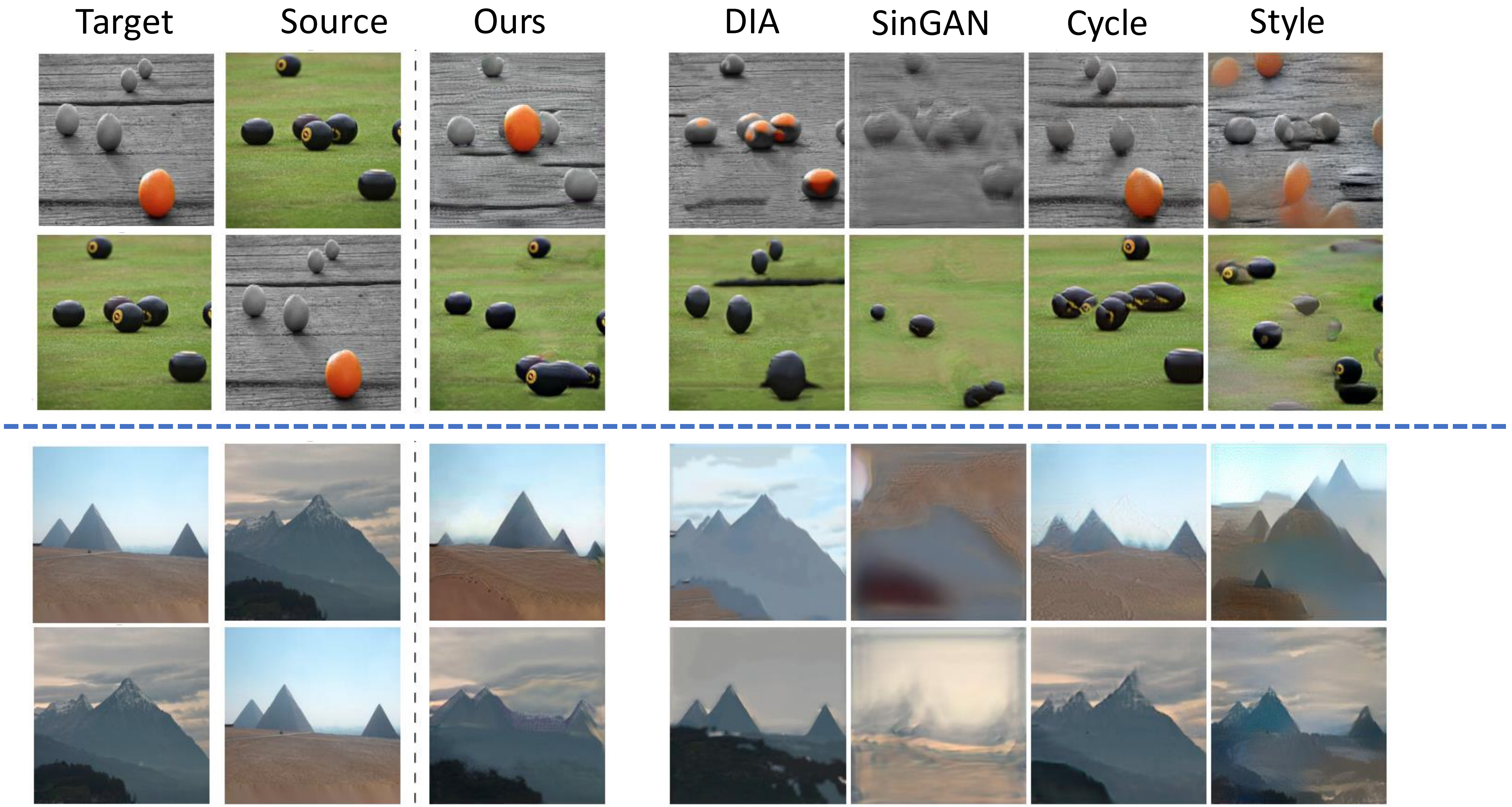


Coarse and Mid Scales: Residual Training



Coarse and Mid Scales: Residual Training

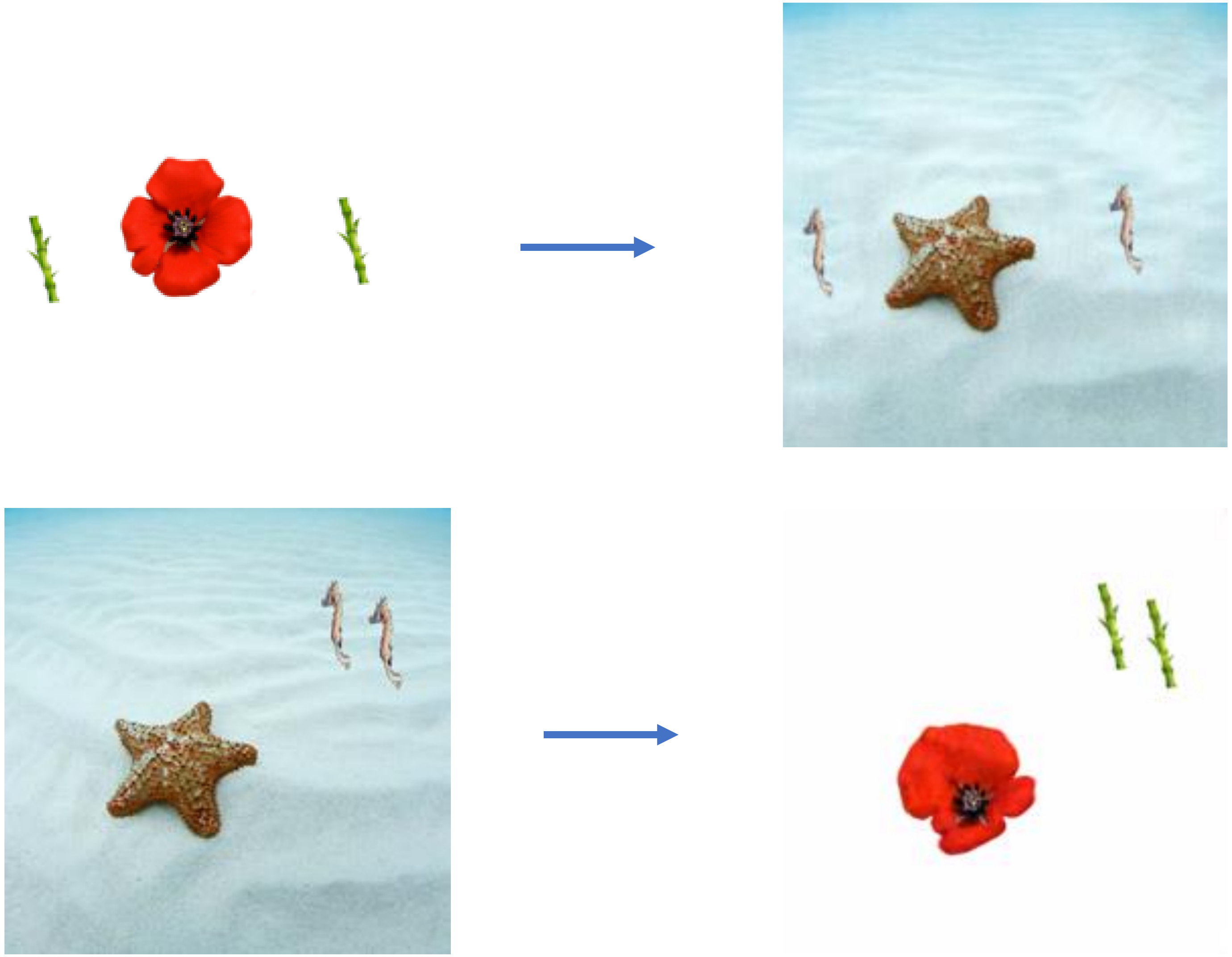



Target
Source
Ours
DIA
SinGAN
Cycle
Style

Multiple Class Types

Input

Output



Paired Generation

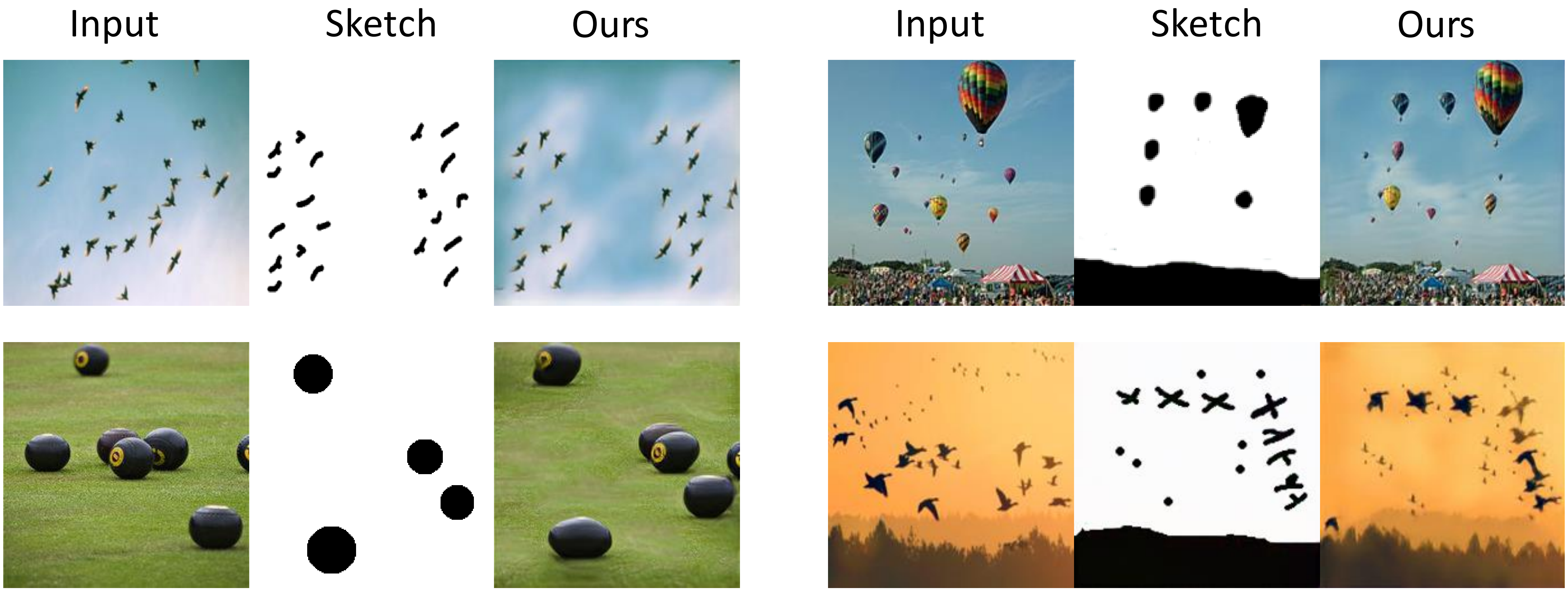
A

~~Un~~conditional

B

~~Un~~conditional

Paint to Image



Video Generation

Permuted AdaIN: Reducing the Bias Towards Global Statistics in Image Classification

O. Nuriel, **S. Benaim**, L. Wolf. Submitted to CVPR 2021.

Reduce bias towards global statistics by swapping the **global statistics** of an image while maintaining its **structure** with probability p, thus improving **image classification tasks**.

Adaptive Instance Normalization

- Let $a \in \mathbb{R}^{C\times H\times W}$ and $b \in \mathbb{R}^{C\times H\times W}$ be the activations of some encoder E applied on images I_a and I_b respectively.
- $\mu_c(a) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} a_{chw}$ (similarly for b)
- $\sigma_c(a) = \sqrt{\sum_{h=1}^{H}\sum_{w=1}^{W} (a_{chw} - \mu_c(a))^2 + \epsilon}$ (similarly for b)
- μ and σ are computed along the **spatial dimension** of a.

$$AdaIN(a,b)_{chw} = \sigma_c(b)\left(\frac{a_{chw} - \mu_c(a)}{\sigma_c(a)}\right) + \mu(b)$$

Adaptive Instance Normalization

$$AdaIN(a,b)_{chw} = \overbrace{\sigma_c(b)}^{\text{Global Statistics}} \underbrace{\left(\frac{a_{chw} - \mu_c(a)}{\sigma_c(a)}\right)}_{\text{Structure}} + \overbrace{\mu(b)}^{\text{Global Statistics}}$$

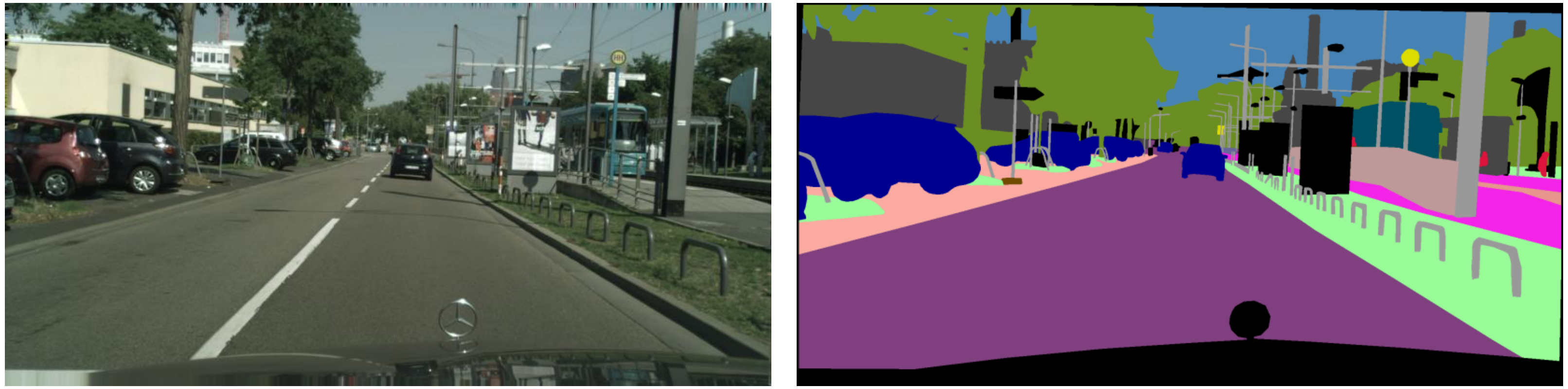
- μ and σ represent the **global statistics** of an image (such as brightness, contrast, lighting, global color changes and global texture)
- **Structure** represents information relating to shape of objects.

Domain Adaptation

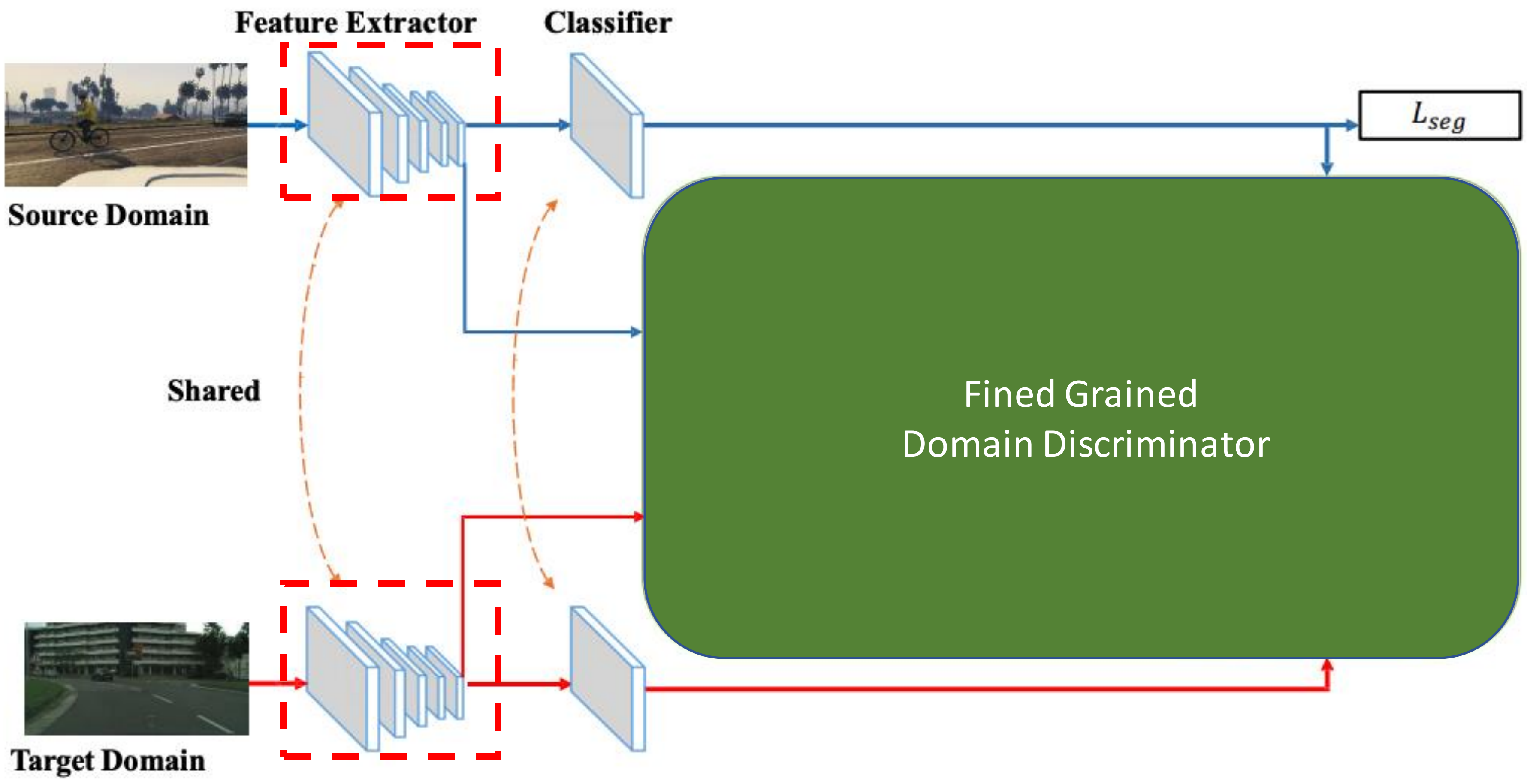
Supervised training on source domain and unsupervised on target domain

Source: GTAV

Target: Cityscapes



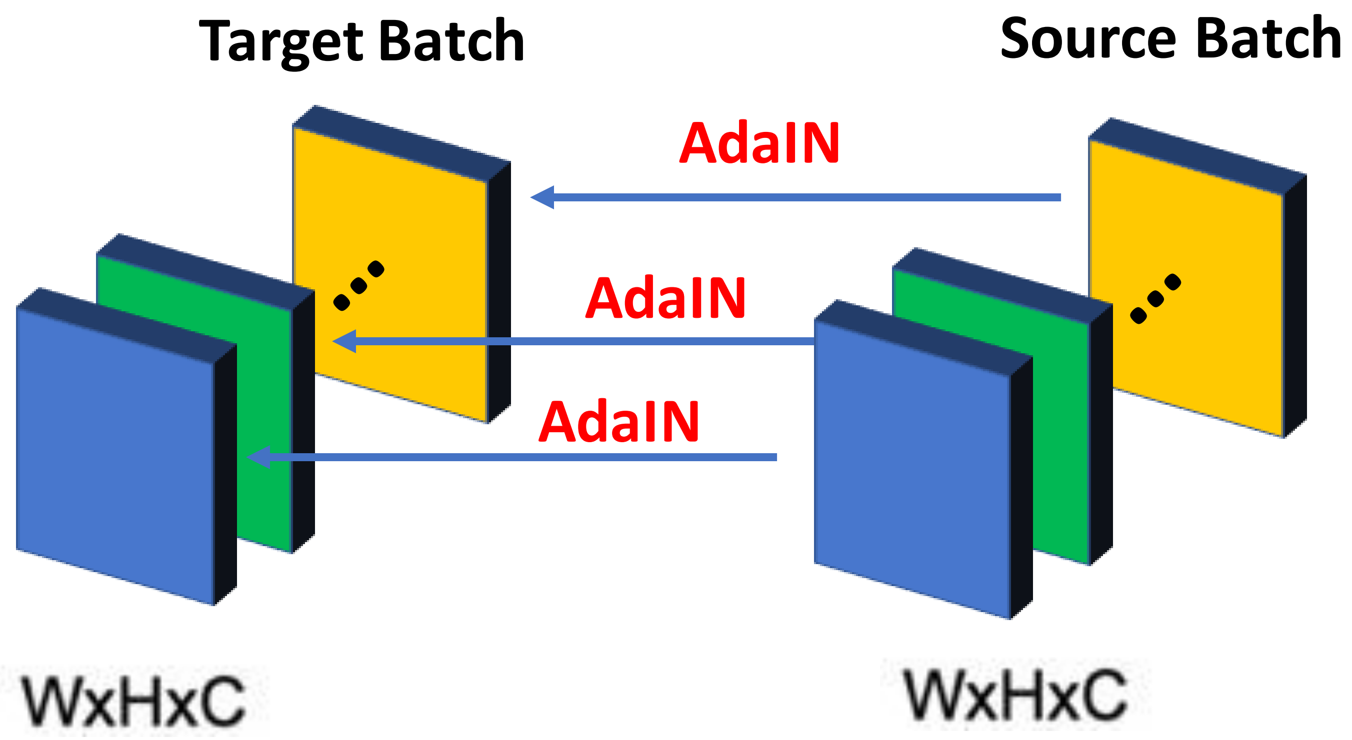
Domain Adaptation



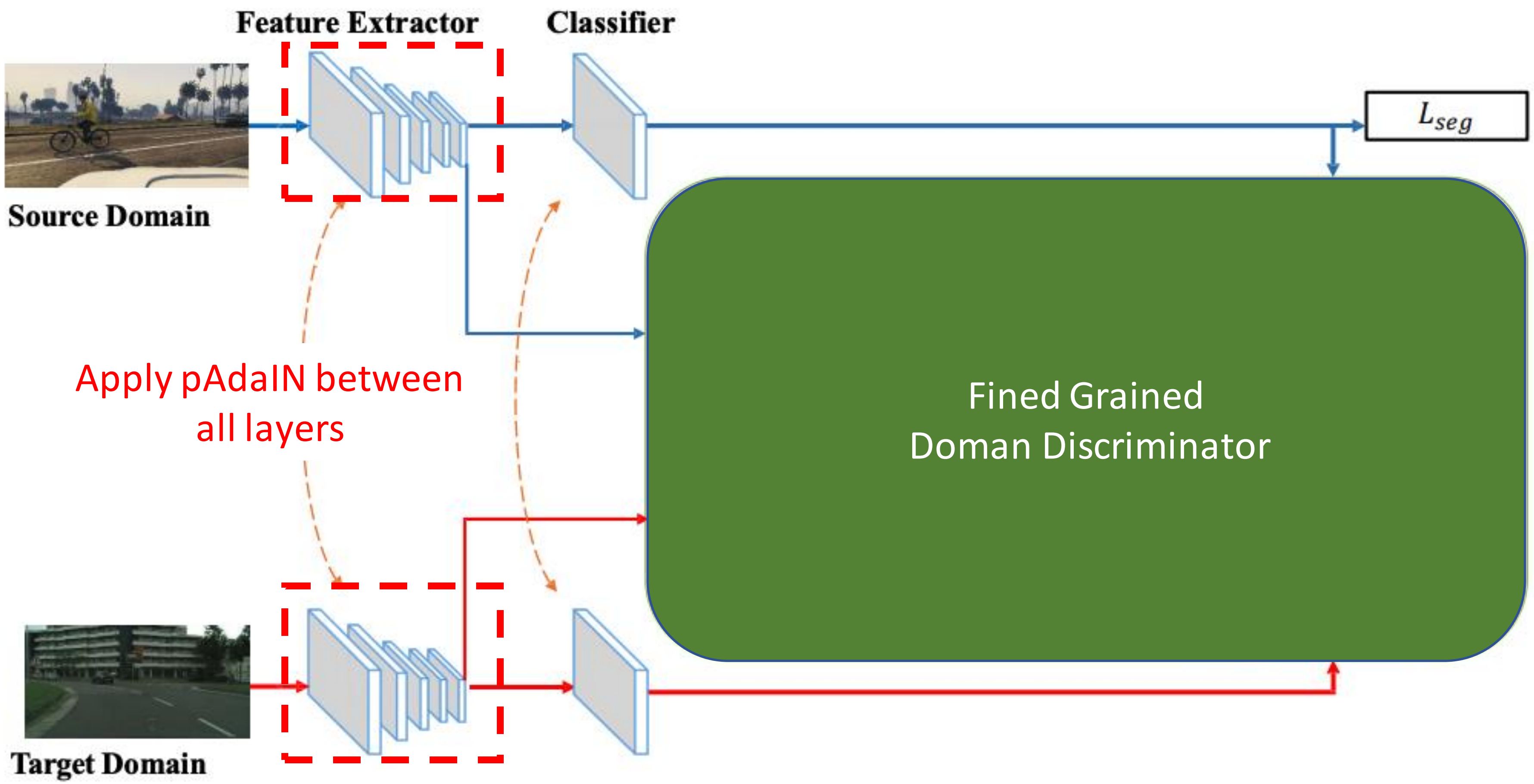
Classes Matter: A Fine-grained Adversarial Approach to Cross-domain Semantic Segmentation. Wang et al., ECCV 2020.

Domain Adaptation

- **Swap global statistics of target features with those of source features** by applying AdaIN with probability p.
- Apply at every layer of the feature extractor.



Domain Adaptation



Classes Matter: A Fine-grained Adversarial Approach to Cross-domain Semantic Segmentation. Wang et al., ECCV 2020.

Domain Adaptation

GTAV to Cityscapes

AdaptSegNet [35]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
SIBAN [28]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
CLAN [29]	87.0	27.1	79.6	27.3	23.3	28.3	**35.5**	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
AdaptPatch [36]	92.3	51.9	82.1	29.2	25.1	24.5	33.8	**33.0**	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5
ADVENT [38]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
FADA [40]	92.5	47.5	85.1	37.6	**32.8**	**33.4**	33.8	18.4	85.3	37.7	83.5	63.2	**39.7**	87.5	32.9	47.8	1.6	34.9	**39.5**	49.2
FADA [40] + pAdaIN	**93.3**	**55.7**	**85.6**	**38.3**	29.6	31.2	34.2	17.8	**86.2**	**41.0**	**88.8**	**65.1**	37.1	**87.6**	**45.9**	**55.1**	15.1	**39.4**	31.1	**51.5**

Domain Adaptation

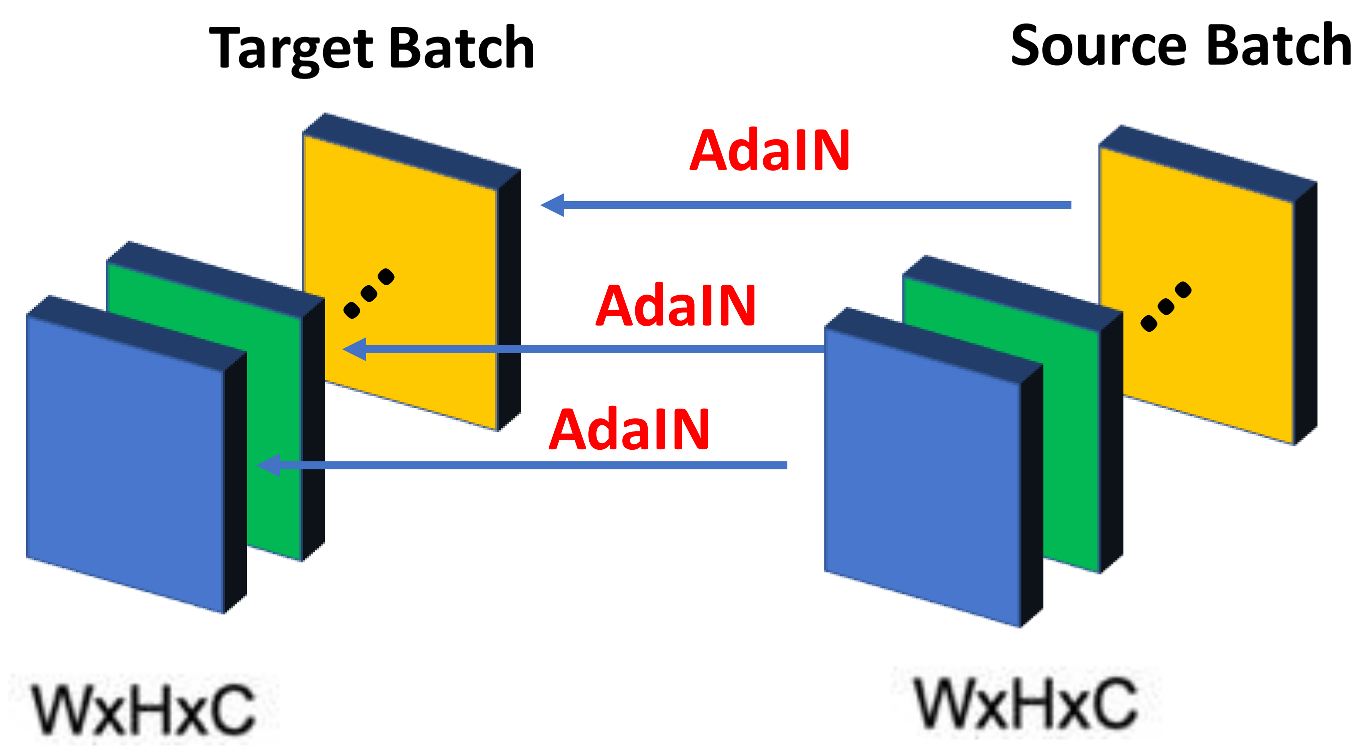


Image Classification

Swap global statistics between every two elements in the batch

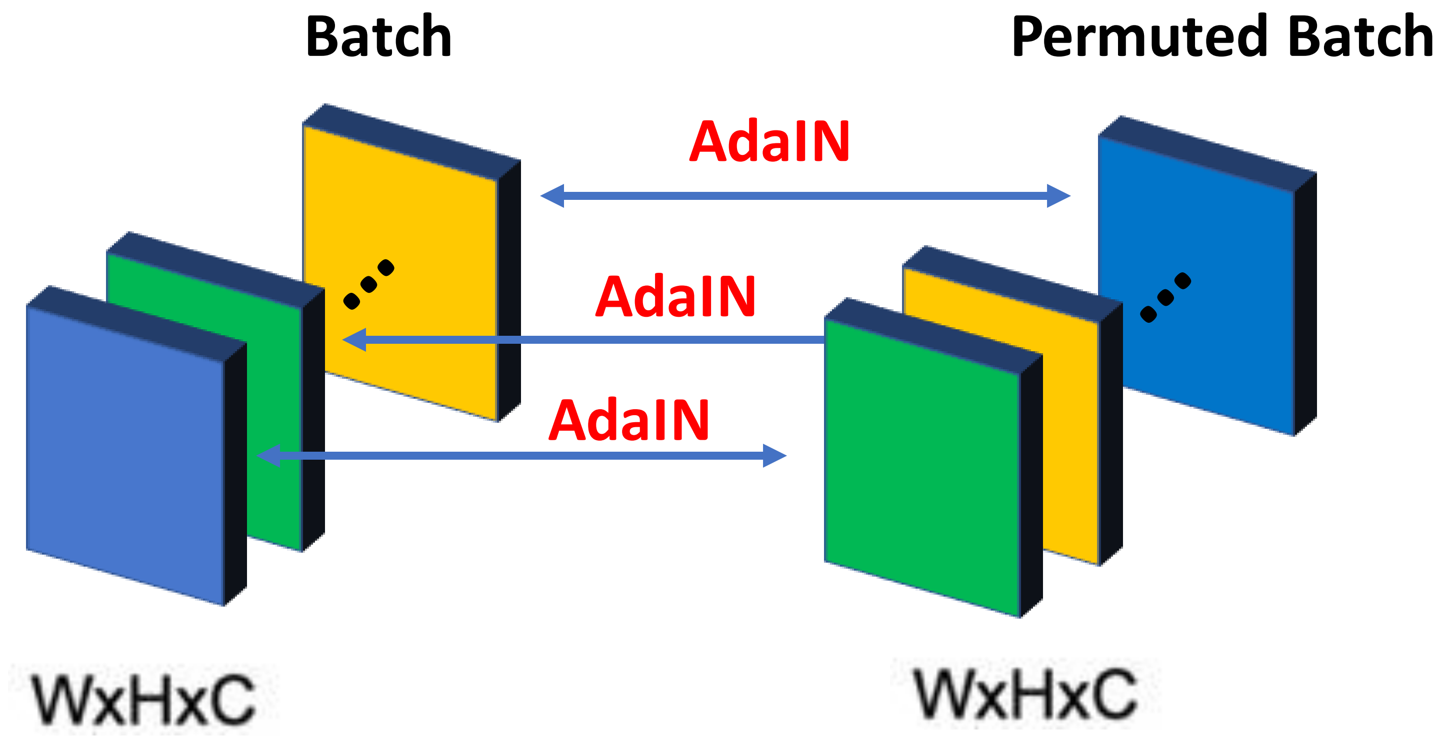


Image Classification

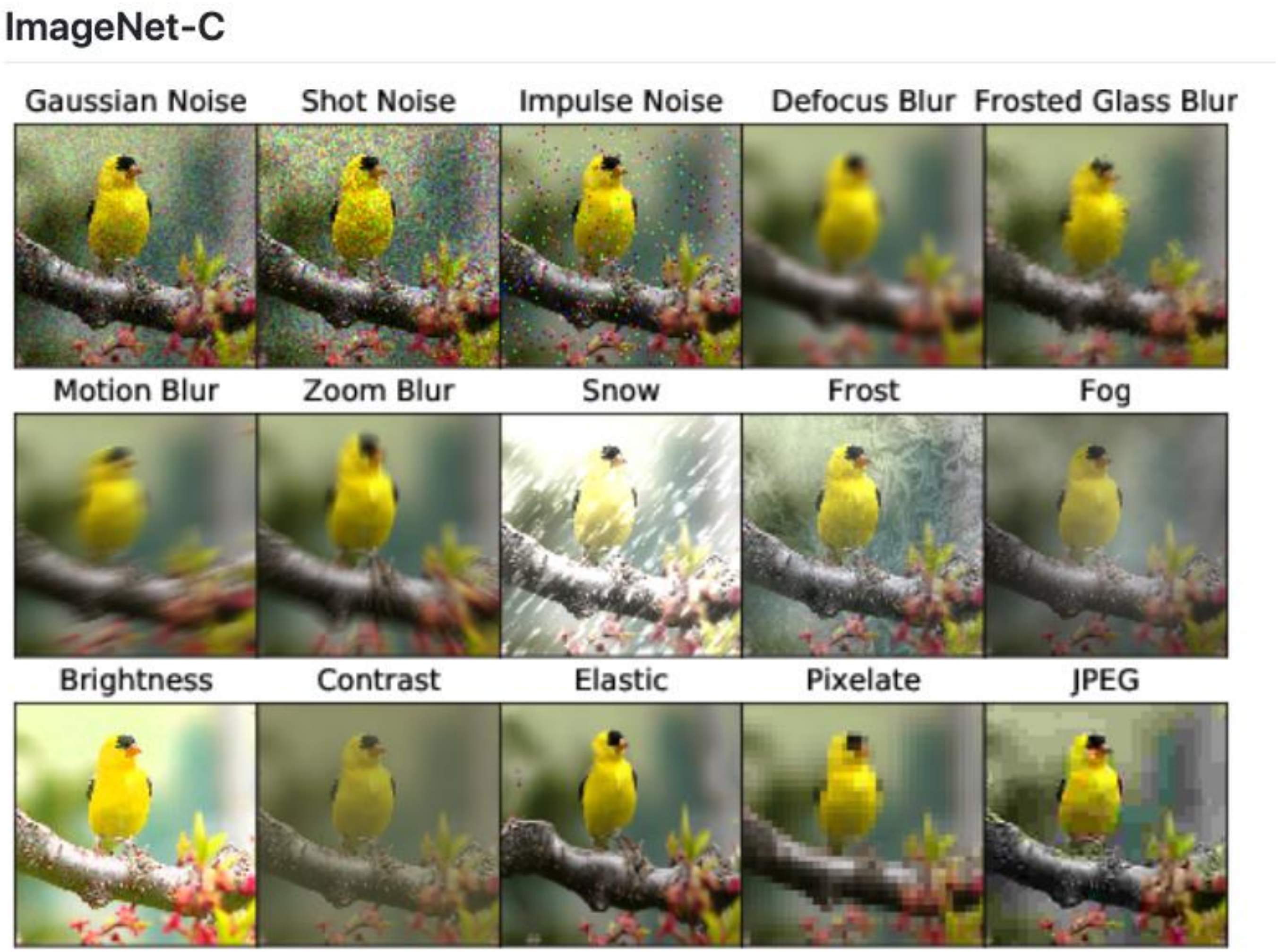
ImageNet

Method	Architecture	Top-1 Accuracy	Top-5 Accuracy
Baseline	ResNet50	77.1	93.63
pAdaIN	ResNet50	**77.7**	**93.93**
Baseline	ResNet101	78.13	93.71
pAdaIN	ResNet101	**78.8**	**94.35**
Baseline	ResNet152	78.31	94.06
pAdaIN	ResNet152	**79.13**	**94.64**

Cifar100

Method	Architecture	CIFAR 100
Baseline	PyramidNet	83.49
pAdaIN	PyramidNet	**84.17**
Baseline	ResNet18	76.13
pAdaIN	ResNet18	**77.82**
Baseline	ResNet50	78.22
pAdaIN	ResNet50	**79.03**

Robustness Towards Corruption



Robustness Towards Corruption

CIFAR100-C

	Baseline	Cutout [8]	Mixup [43]	CutMix [43]	Auto-Augment [7]	Adversarial Training [30]	Augmix [18]	pAdaIN+ Augmix
DenseNet-BC	59.3	59.6	55.4	59.2	53.9	55.2	38.9	**37.5**
ResNext-29	53.4	54.6	51.4	54.1	51.3	54.4	34.4	**31.6**

Category Wise Breakdown

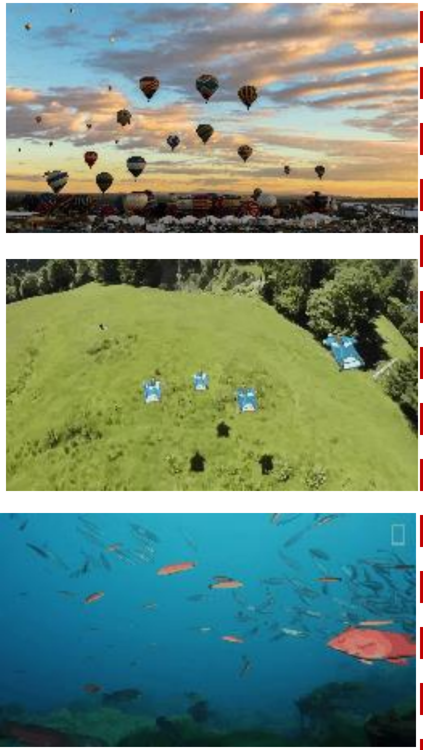
Dataset	Network	Architecture	E	mCE	Noise			Blur				Weather				Digital			
					Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
INet-C	Baseline	ResNet50	22.9	76.7	80	82	83	75	89	78	80	78	75	66	57	71	85	77	77
INet-C	pAdaIN	ResNet50	**22.3**	**72.8**	**78**	**79**	**81**	**70**	**87**	**74**	**76**	**74**	**71**	**64**	**55**	**65**	**82**	**66**	**71**
C100-C	Augmix [18]	DenseNet-BC	24.2	38.9	60	51	41	27	55	31	29	36	39	35	28	37	33	39	41
C100-C	Augmix+pAdaIN	DenseNet-BC	**22.2**	**37.5**	**58**	**49**	**40**	**26**	**54**	**30**	**28**	**35**	**38**	**33**	**25**	**36**	**32**	**37**	**40**
C100-C	Augmix [18]	ResNext-29	21.0	34.4	**56**	**48**	32	23	**49**	27	25	32	35	32	24	32	30	34	37
C100-C	Augmix+pAdaIN	ResNext-29	**17.3**	**31.6**	58	**48**	**24**	**20**	54	**23**	**21**	**28**	**30**	**25**	**19**	**27**	**27**	**33**	**36**

Videos?

Hierarchical Patch VAE-GAN: Generating Diverse Videos from a **Single Sample**

S. Gur*, **S. Benaim***, L. Wolf. NeurIPS 2020 (*Equal contribution)

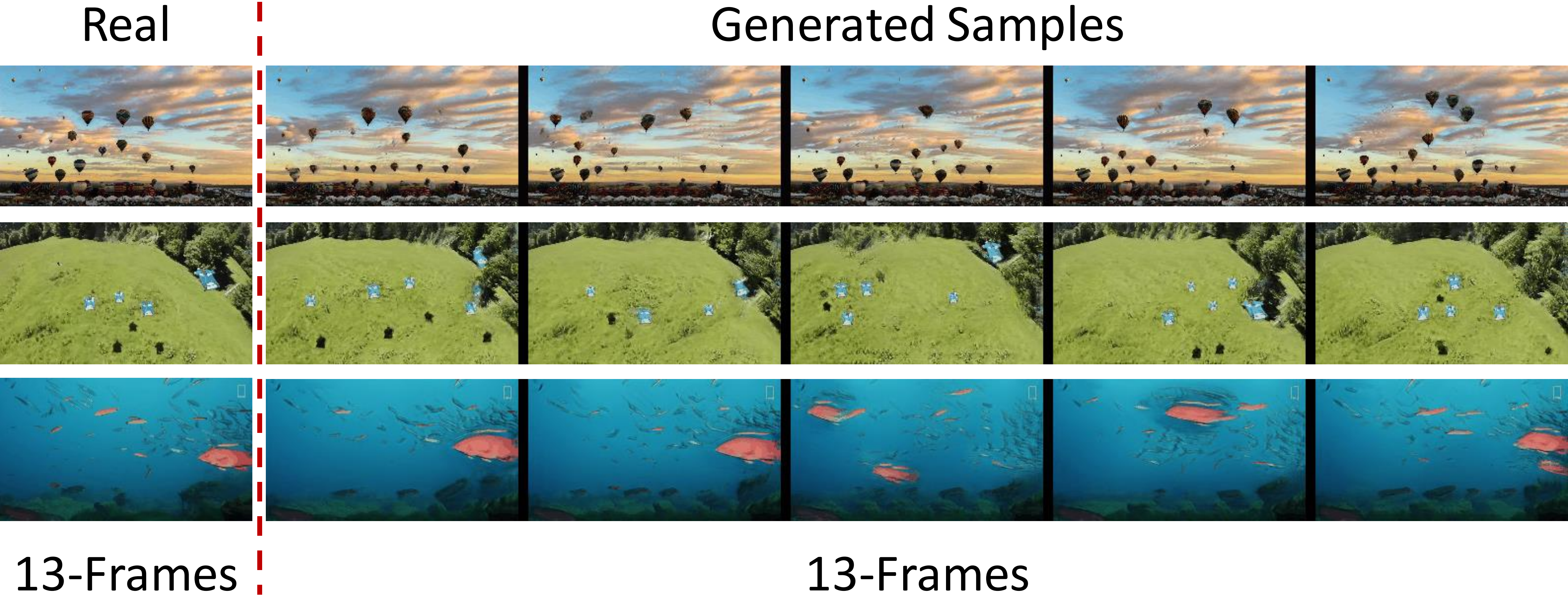
Real



13-Frames

Hierarchical Patch VAE-GAN:
Generating Diverse Videos from a Single Sample

S. Gur*, **S. Benaim***, L. Wolf. NeurIPS 2020 (*Equal contribution)



Extending 2D to 3D

Real | Ours

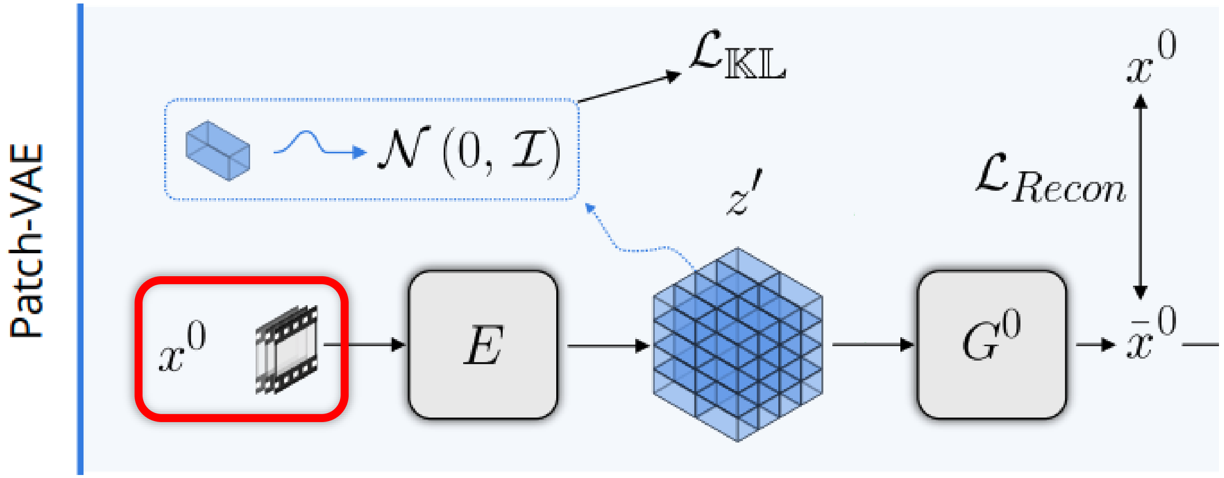
Real | SinGAN [1] + 3D Convolution

Real | ConSinGAN [2] + 3D Convolution

[1] “SinGAN: Learning a Generative Model from a Single Natural Image”, Shaham et al., ICCV 2019
[2] “Improved Techniques for Training Single-Image GANs”, Hinz et al., arXiv 2020

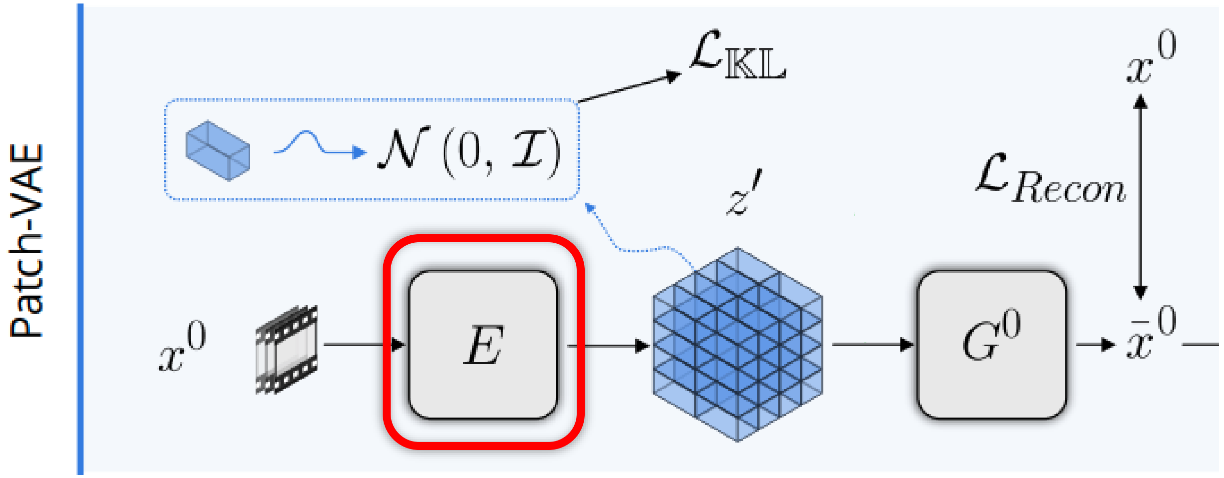
Proposed Approach: Patch VAE

Input video - x^0



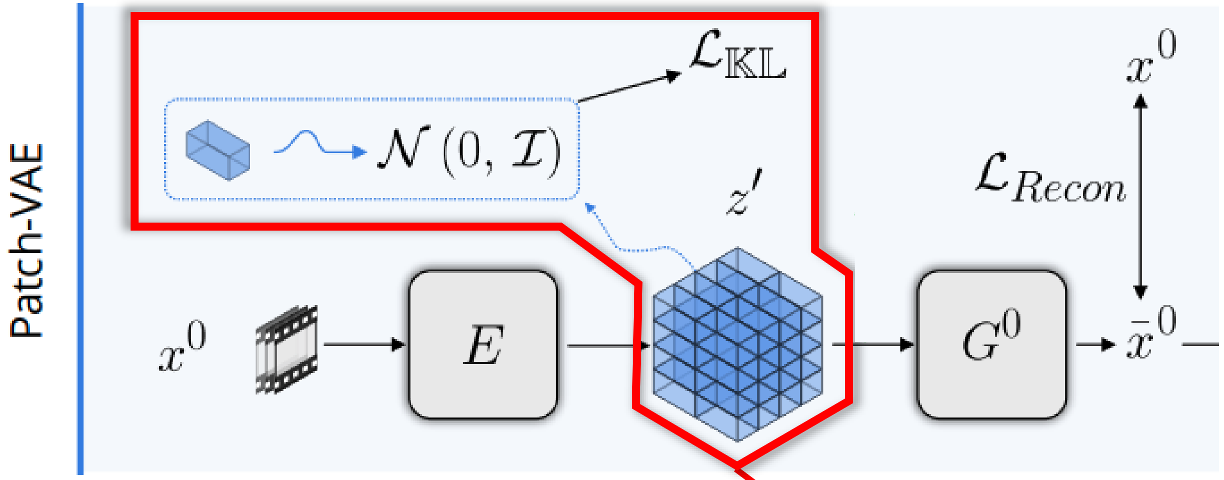
Proposed Approach: Patch VAE

Encoder – $\mathrm{E}(x^0)$

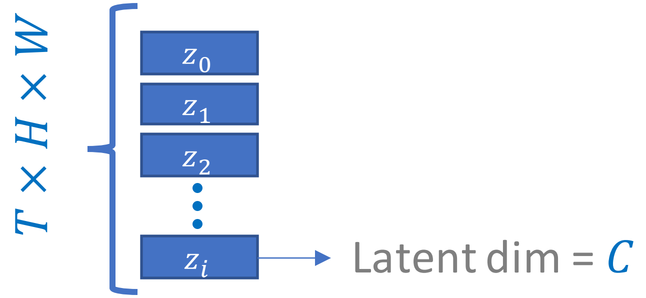


Proposed Approach: Patch VAE

KL Loss – z'

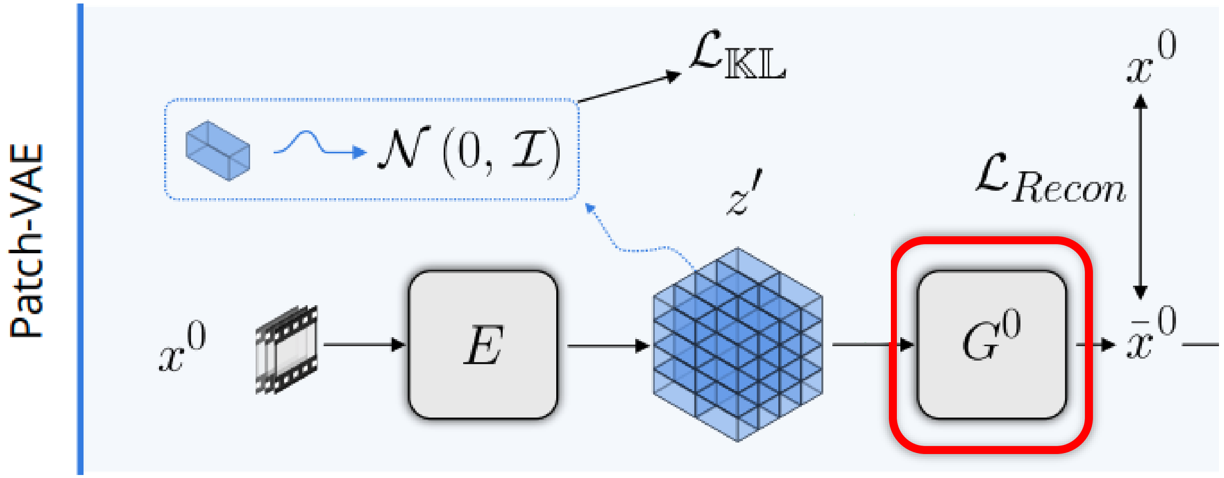


Each feature z_i , $i = [1 \dots K], K = T \times H \times W$,
in the latent space is associated with a patch ω_i



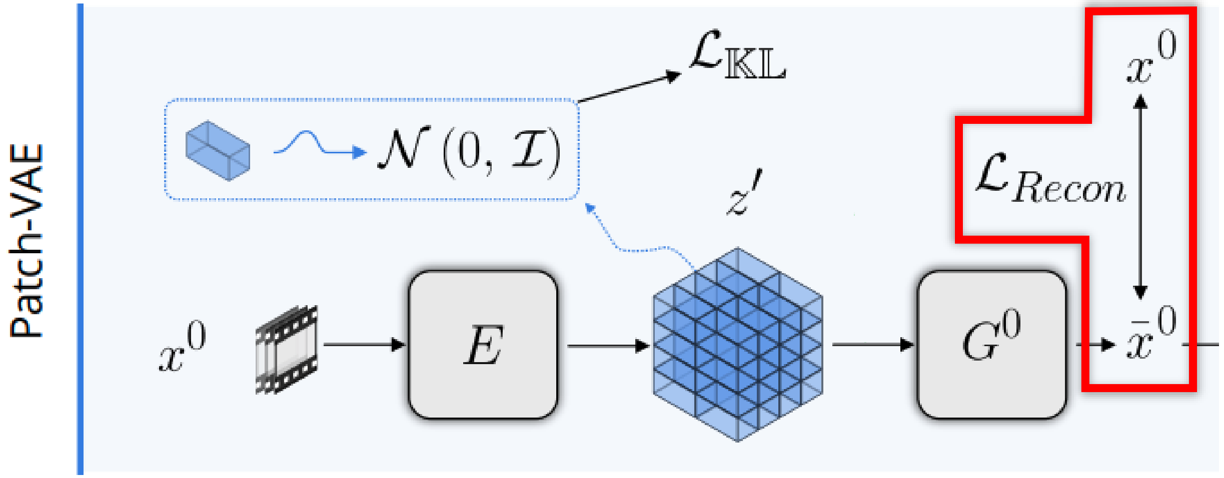
Proposed Approach: Patch VAE

Decoder - $\bar{x}^0$



Proposed Approach: Patch VAE

Reconstruction loss



Proposed Approach: Hierarchical Patch VAE

Coarsest scale:
Low resolution
and frame rate

x^0 (Real)
$\bar{x}^0$ (Generated)

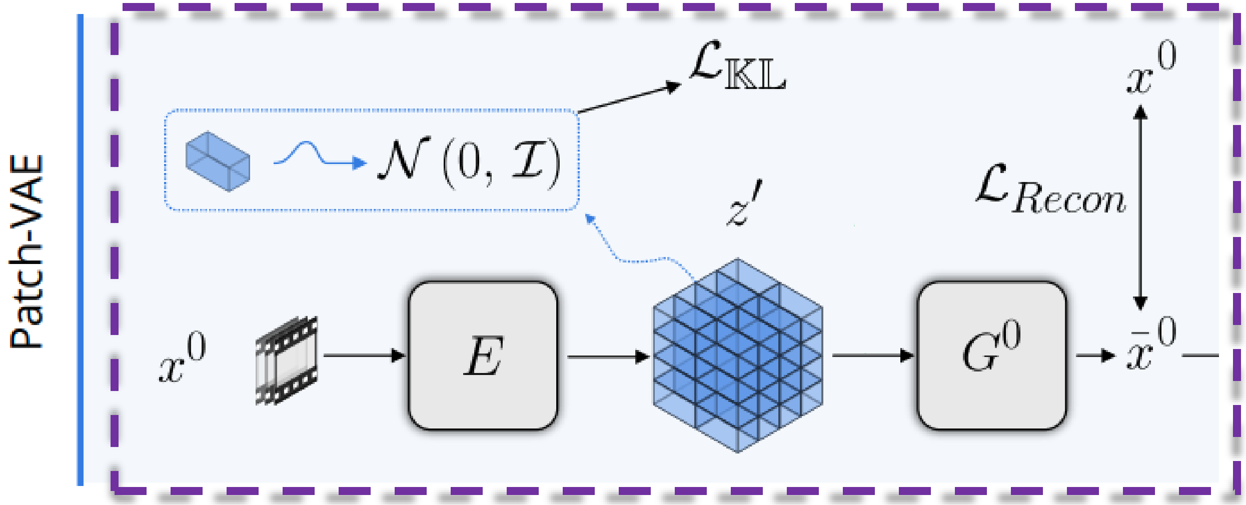
LEVEL $= 0$

Finest scale:
High resolution
and frame rate

x^N (Real)
$\bar{x}^N$ (Generated)

LEVEL $= N$

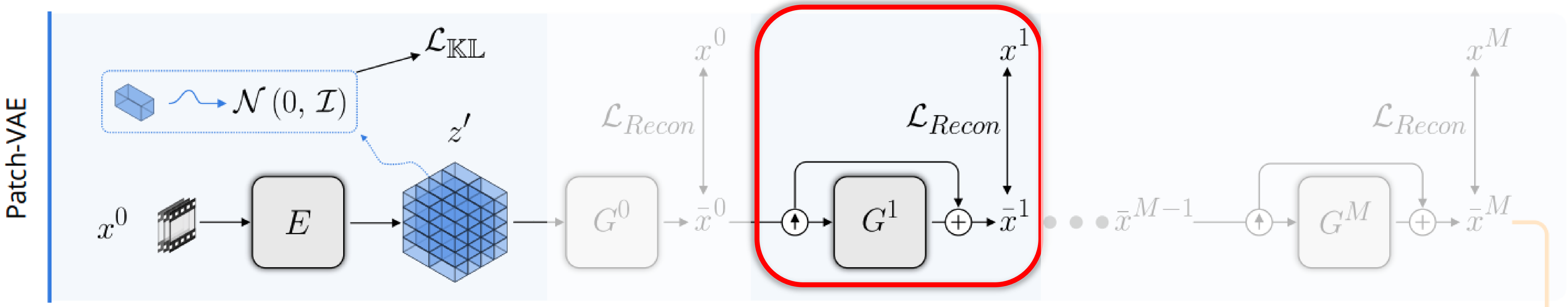
Proposed Approach: Hierarchical Patch VAE



LEVEL $= 0$

Proposed Approach: Hierarchical Patch VAE

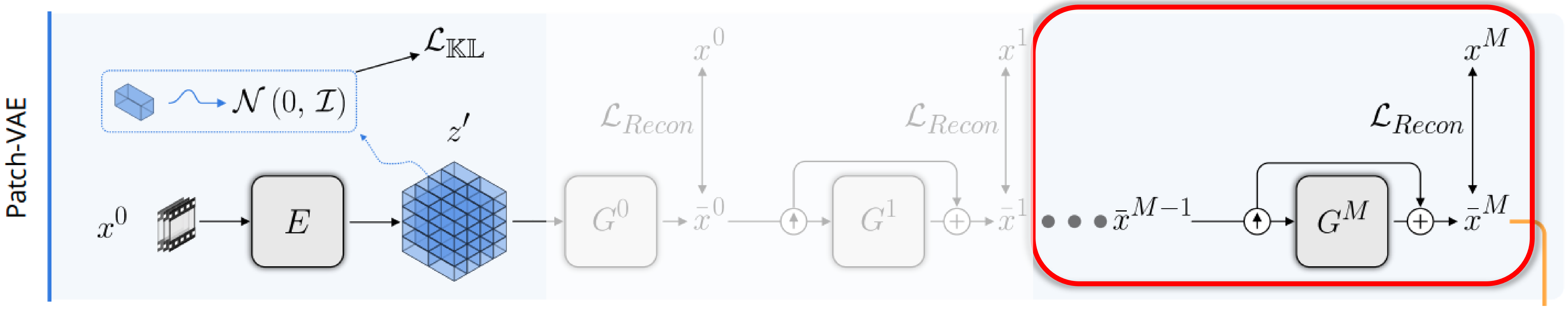
Up-sampling block - $\bar{x}^1$



LEVEL = 1

Proposed Approach: Hierarchical Patch VAE

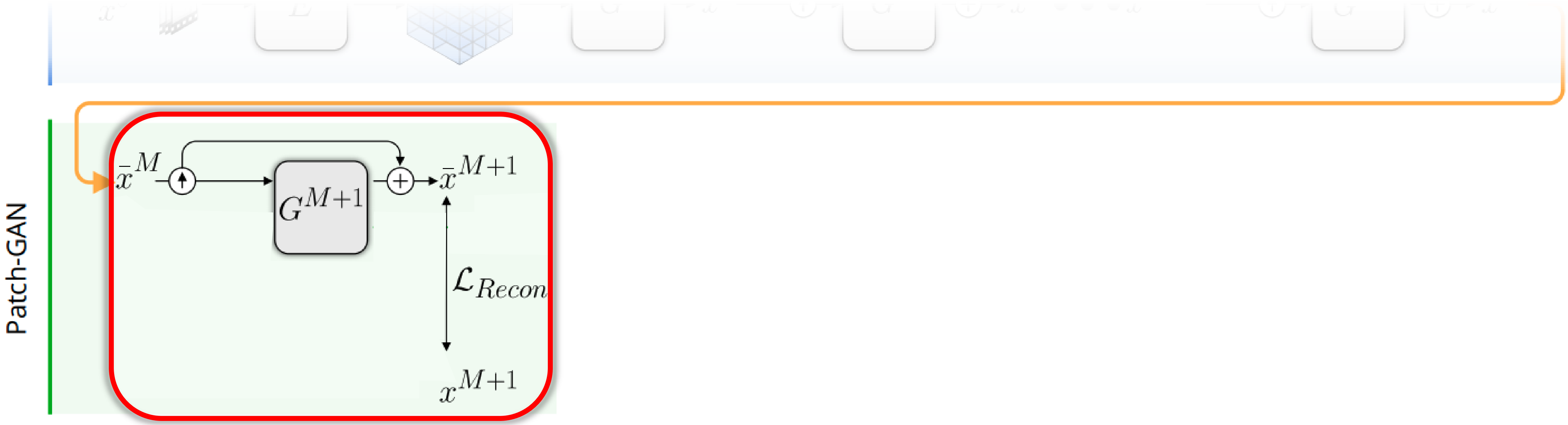
Hierarchical up-sampling up to $\bar{x}^M$



LEVEL $\leq M$

Proposed Approach: Hierarchical Patch VAE GAN

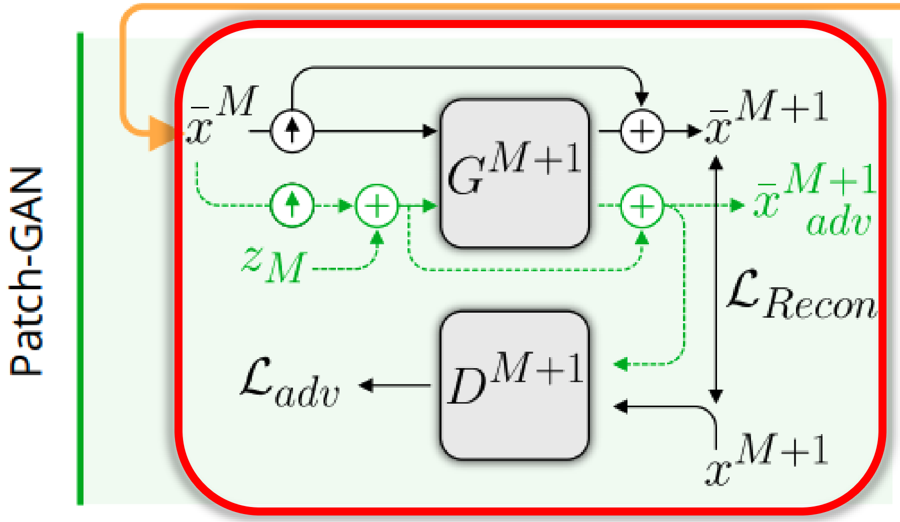
Up-sampling block $\bar{x}^{M+1}$



LEVEL $= M + 1$

Proposed Approach: Hierarchical Patch VAE GAN

Adversarial training

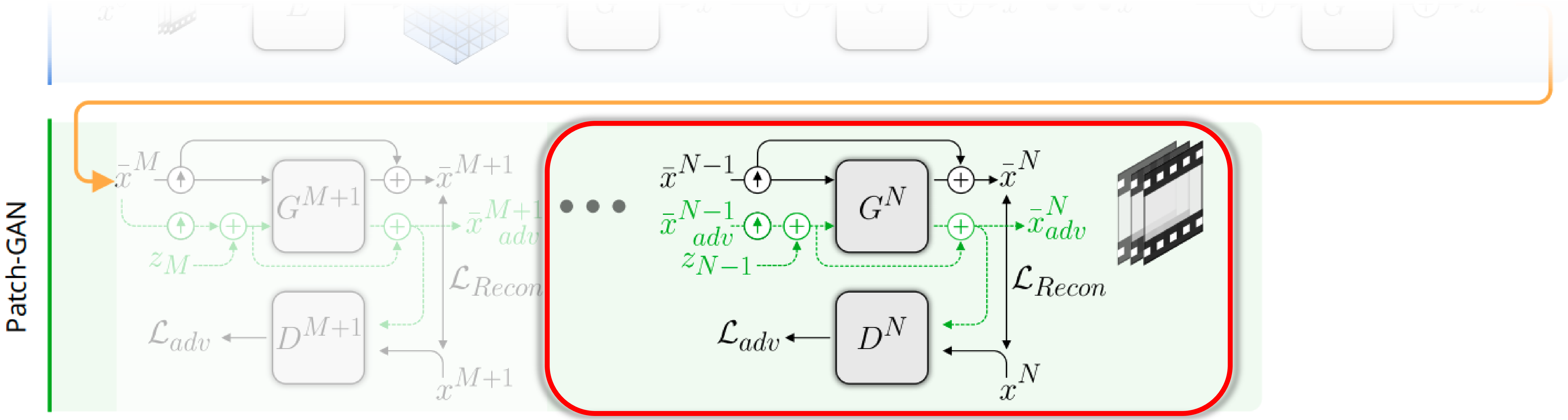


Added noise z_M

LEVEL $= M + 1$

Proposed Approach: Hierarchical Patch VAE GAN

Hierarchical up-sampling up to final resolution $\bar{x}^N$



$M + 1 <$ LEVEL $\leq N$

Effect of Number of patch-VAE levels

Training Video

9 Levels Total

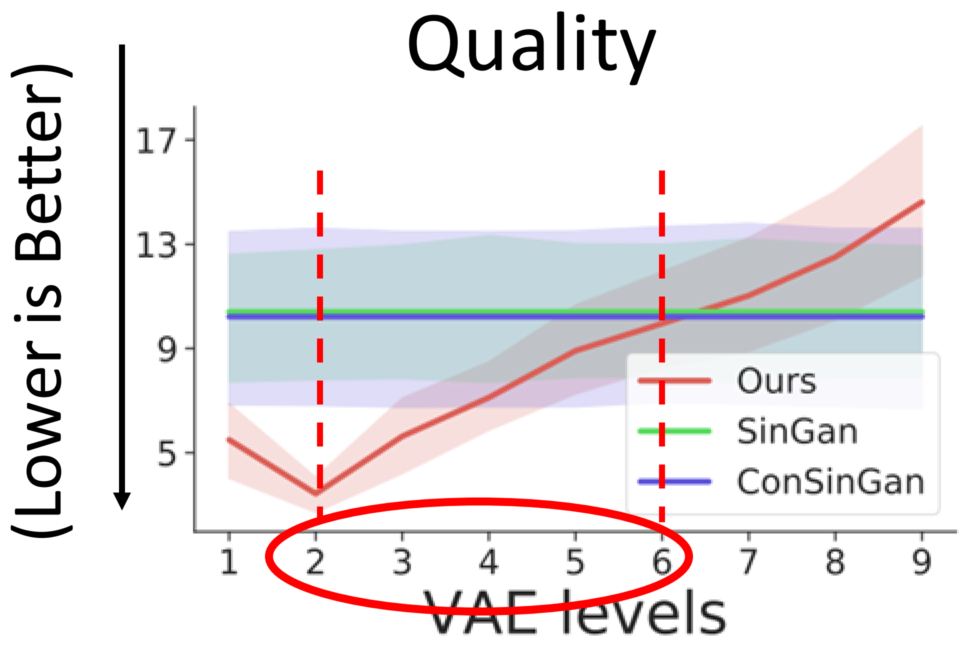
1 p-VAE – 8 p-GAN

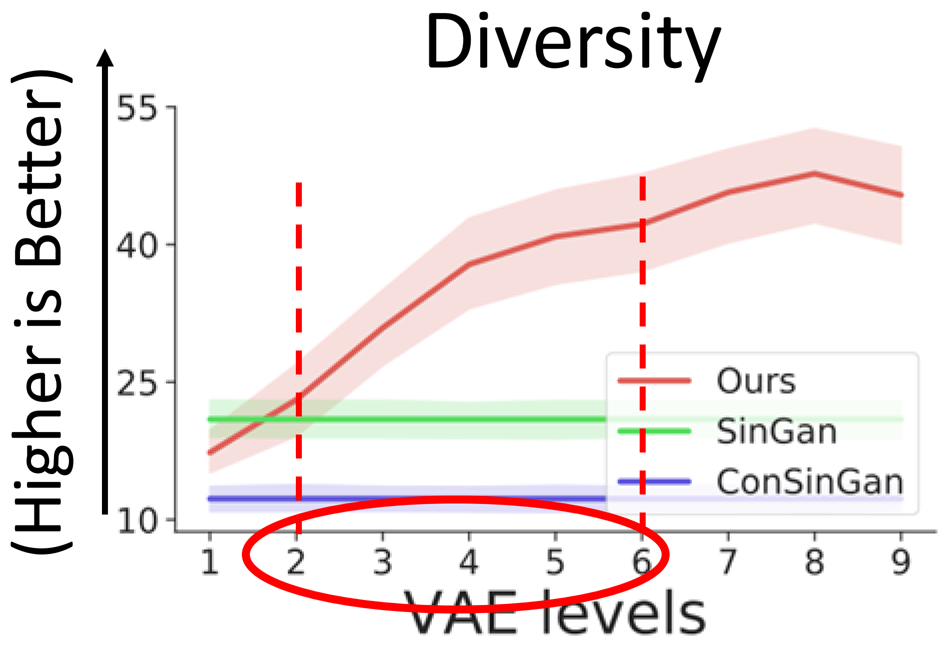
8 p-VAE – 1 p-GAN

3 p-VAE – 6 p-GAN

Effect of Number of patch-VAE levels

Total of 9 layers





SpeedNet: **Learning the Speediness in Videos**

S. Benaim, A. Ephrat, O. Lang, I. Mosseri, W. T. Freeman, M. Rubinstein, M. Irani, T. Dekel.
CVPR 2020.

Slower

Normal speed

Faster

Automatically predict "speediness"

Uniform Speed Up (2x)

Adaptive speed up (2x)

Other Applications:

- Self-supervised action recognition
- Video retrieval

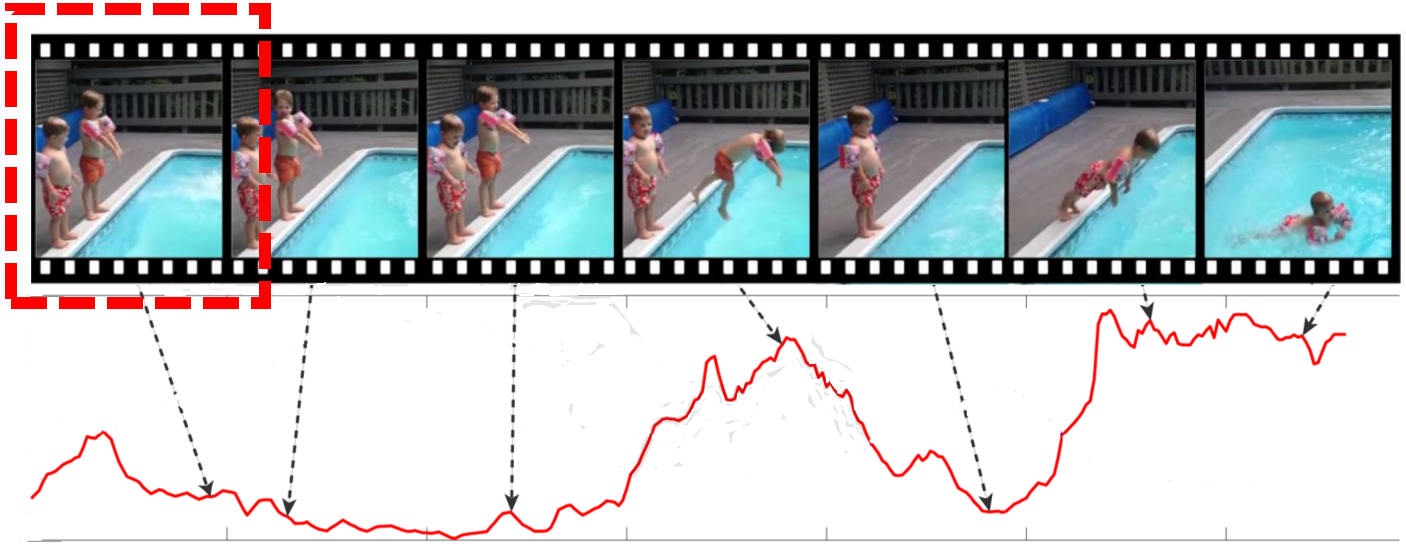
SpeedNet

Self-supervised training

Sped Up

Input video

Inference on full **sped-up** video

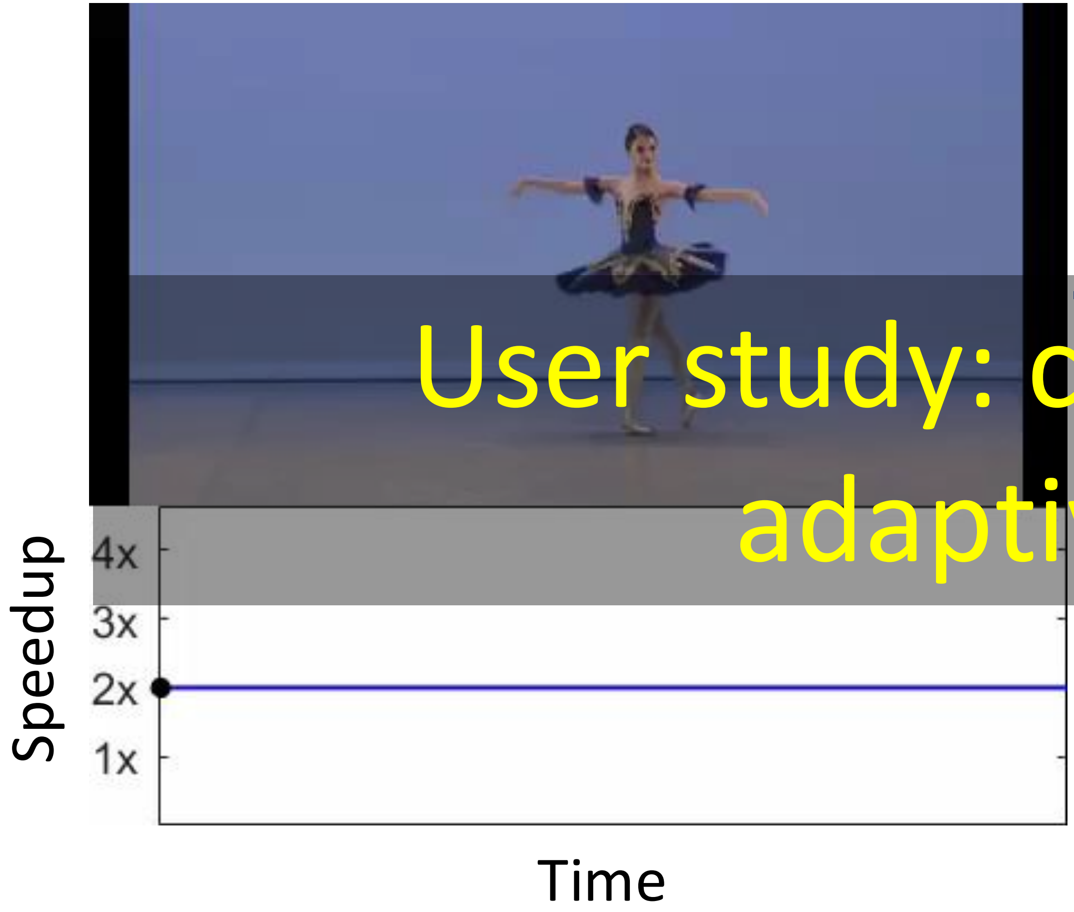


Sped-up

Normal speed

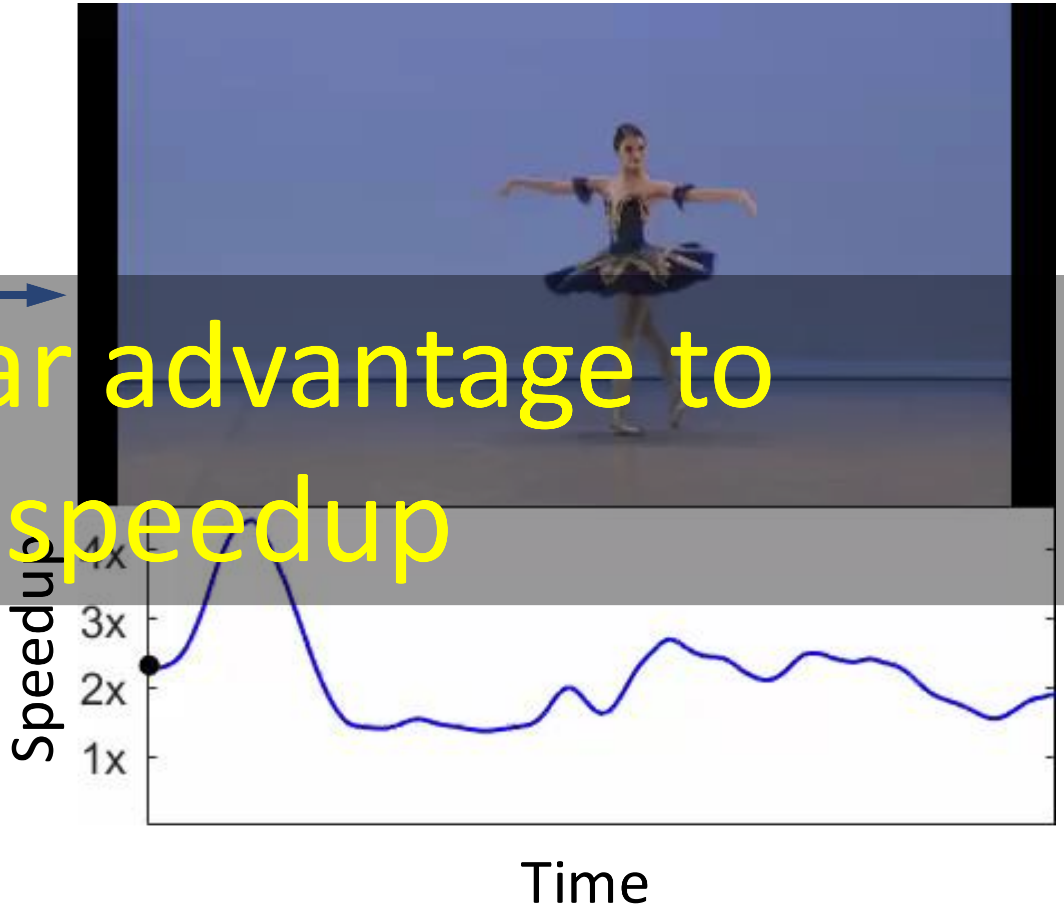
Adaptive video speedup

Total time = $\frac{1}{2}$ input time



Uniform Speedup

Total time = $\frac{1}{2}$ input time

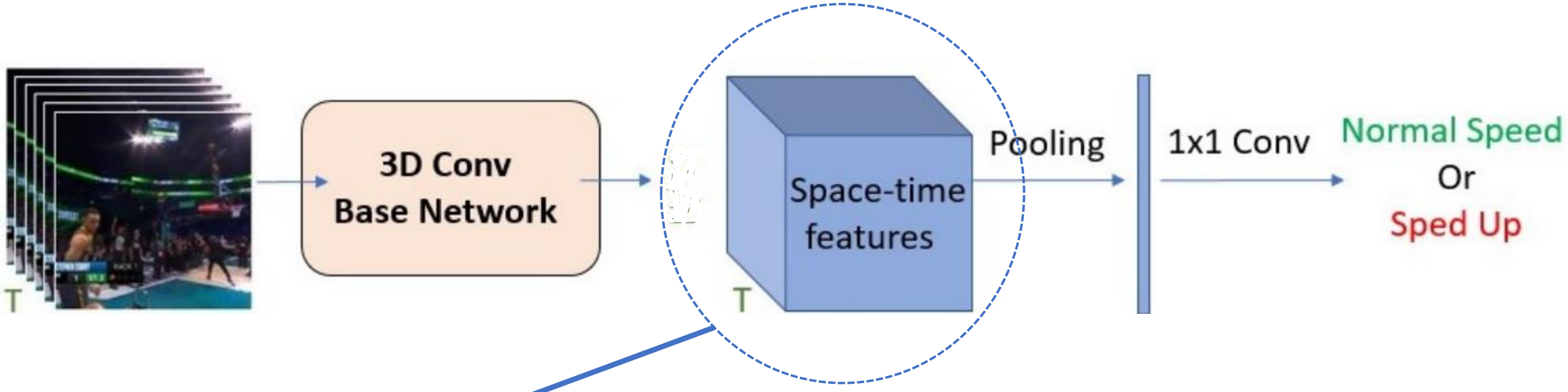


Adaptive Speedup (ours)

User study: clear advantage to adaptive speedup

Other self supervised tasks

Train SpeedNet

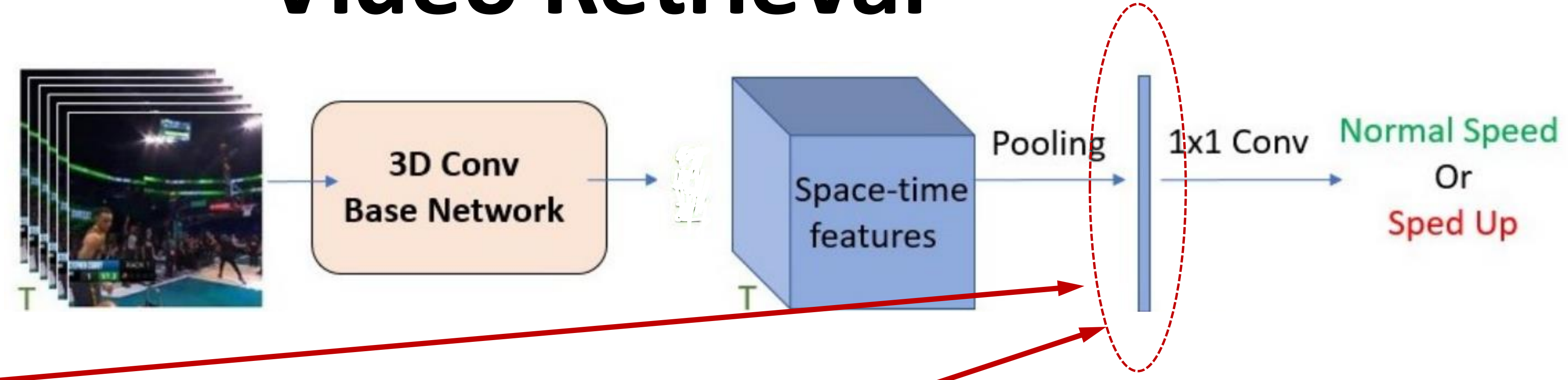


Self Supervised Action Recognition

Initialization		Supervised accuracy	
Method	**Architecture**	**UCF101**	**HMDB51**
Random init	S3D-G	73.8	46.4
ImageNet inflated	S3D-G	86.6	57.7
Kinetics supervised	S3D-G	96.8	74.5
CubicPuzzle [19]	3D-ResNet18	65.8	33.7
Order [40]	R(2+1)D	72.4	30.9
DPC [13]	3D-ResNet34	75.7	35.7
AoT [38]	T-CAM	79.4	-
SpeedNet (Ours)	S3D-G	**81.1**	**48.8**
Random init	I3D	47.9	29.6
SpeedNet (Ours)	I3D	66.7	43.7

Other self supervised tasks: Video Retrieval

Train SpeedNet

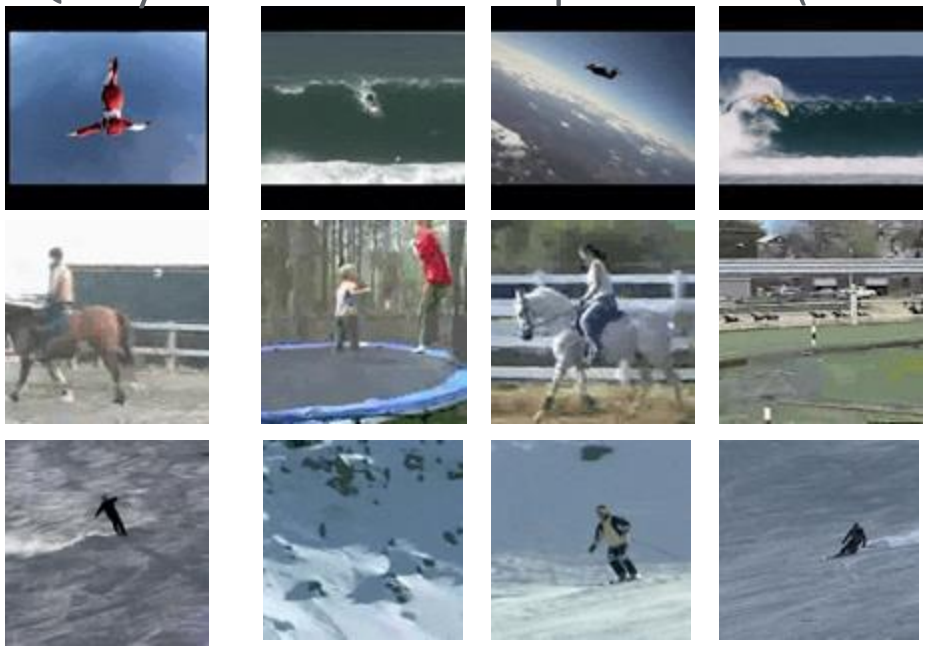


Query

Retrieved top-3 results (Within)

Query

Retrieved top-3 results (Across)



“Memory Eleven”: An artistic video by Bill Newsinger:
https://www.youtube.com/watch?v=djylS0Wi_Io

Spatio-Temporal Visualizations

blue/green =
normal speed

yellow/orange =
slowed down

Conclusion

- Going beyond texture and style manipulation
- Structure manipulating in images:
  - Fully supervised (pix2pix, spade): expensive supervision of segmentation masks
  - Two unpaired domains
  - A single image pair
  - Downstream tasks: image classification and domain adaptation
- Structure manipulation in videos:
  - Single video: novel videos capturing similar object structure
  - Speeding up videos “gracefully” using “speed” as supervision
- Next?
  - Structure manipulation in 3D
  - Videos from multiple scenes
  - “Functional relationships”

Thank You! Questions?